TEXT EXTRACTION UND SEMANTIC WEB AM BEISPIEL VON MARKTSTUDIEN

Egy Rizky Utama RIDWAN

Diplomarbeit an der Fachhochschule Köln, Campus Gummersbach, Fakultät für Informatik und Ingenieurwissenschaften

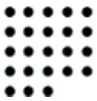
Text Extraction und *Semantic Web* am Beispiel von Marktstudien

Für das Marktstudienportal www.markt-studie.de und seine englisch- und spanischsprachigen Pendants www.reports-research.com bzw. www.estudio-mercado.es soll ein Weg gefunden, die Kategorisierung von Marktstudien automatisch zu unterstützen, so dass der Zeitaufwand für die Marktstudienimporte mitsamt der Kategorienzuordnung verringert wird. Zusätzlich dazu sollen die zu kategorisierenden Martkstudien in einer Web Ontologie abgespeichert werden.

Abgabe: 08. Juli 2010
Letzte Korrektur: 07. Juli 2010
Note:

Der Autor:

Egy Rizky Utama Ridwan r.ridwan@markt-studie.de



Fachhochschule Köln
Cologne University of Applied Sciences

Text Extraction und *Semantic Web* am Beispiel von Marktstudien

Am Institut für Informatik
an der Fachhochschule Köln
Campus Gummersbach

im Studiengang Allgemeine Informatik

zur
Erlangung des Grades Diplom-Informatiker (FH)
eingereichte

D i p l o m a r b e i t

vorgelegt von

Egy Rizky Utama Ridwan

Erste Prüferin:	Prof. Dr. Heide Faeskorn-Woyke
Zweite Prüferin:	Prof. Dr. Edda Leopold

Gummersbach, im Juli 2010

Für meine Mutter,
möge sie von oben herabscheinen
und alles hell erleuchten

Abstract – Deutsch

Das Erstellen von Textzusammenfassungen ist bei Recherchearbeiten die gängigste Praxis, um einem Text seine Kernaussage zu entnehmen. Aus einer Zusammenfassung leitet man die essenzielle Information ab, mit dem Ziel, einen Text einem Themengebiet zuzuordnen. Dem Benutzer hilft hierbei ein software-gestütztes System zur Textzusammenfassung.

Texte beinhalten aus Sicht eines Computersystems eine Aneinanderreihung von Wörtern bzw. Sätzen und besitzen demnach keine feste Struktur. Aus den unstrukturierten Daten im Text müssen Wörter extrahiert werden, die den Kerninhalt eines Textes widerspiegeln. Auf Basis der extrahierten Wörter erfolgt die endgültige Zusammenfassung und anschließend die semantische Auszeichnung des Gesamttextes, was der Themengebietszuordnung entspricht.

Bewährte Methoden für die Textzusammenfassung sind das statistische Verfahren und das sogenannte Signalwort Verfahren. Grundlage dafür sind die theoretischen Arbeiten von H. P. Luhn und Edmundson. Als weitere Hilfsmittel dienen statistische Methoden aus dem Text Mining. Für die Zuordnung des Textes zu einem Themengebiet eignen sich die Semantic Web Standards des W3C.

Der auf Basis der Theorien entwickelte Lösungsweg dient als Standardverfahren für eine software-gestützte Textzusammenfassung. Durch die definierten Standards ist die Software in der Lage, jegliche Textarten aus beliebiger Quelle einzulesen, zusammenzufassen und semantisch auszuzeichnen.

Benutzer, deren Aufgabe im Unternehmen darin besteht, Recherchearbeiten zu betreiben, verwenden diese software-gestützte Textzusammenfassung. Durch diese Unterstützung spart der Benutzer bei einer Zusammenfassung von mehreren Textdaten Zeit und Aufwand, da der Vorgang durch die Software automatisch abläuft. Aus Sicht eines Unternehmens liegt das Hauptaugenmerk auf der schnellen Informationsgewinnung aus Texten, was essenziell für eine Weiterverarbeitung der Textdaten ist.

Abstract – Englisch

Creating a text summary is a common practice in the research field for withdrawing the main quintessence of the text. From the text summary the main information will be derived in order to assign the text to a topic. A software based solution for Text-Summarization supports the user to achieve the goal.

From the computersystems' point of view, texts contain a series of words and sentences respectively, thus texts do not have a certain structure, which means texts are more or less unstructured. Words have to be extracted from this unstructured data, which reflects the core content of the text. Based on this extracted words follows the summarization and after that the semantic annotation of the text, which is simmilar to the assignment of the text to a topic.

The two main methods for Text-Summarization are the statististical procedure and the so-called Cue-Word-Method. The basic principles for these methods are based on the theoretical works of H. P. Luhn and Edmundson. Additionaly statistical methods from the Text Mining field also support the Text-Summarization. The proposals of the Semantic Web standards by the W3C will be used to assign a text to a topic.

The developed solution, which is based on the theoretical proposals is used as a standard procedure for a software based Text-Summarization. By defining these standards the software is capable to read, summarize and annotate any form of text from any source.

Users who are specialized in conducting text summaries in a company use this software-based Text-Summarization. By supporting the user on the summarization process, time and effort are saved, since the Text-Summarization procedure is automated. From the companies' point of view the main focus lies on the fast and rapid information extraction of texts, which is essential for further data processing.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungs- und Symbolverzeichnis

API	Application Programming Interface
bzgl.	bezüglich
bzw.	beziehungsweise
CSV	Comma-seperated Values
d. h.	das heißt
DL	Description Logic
ggf.	gegebenenfalls
HKL	Häufigkeitsklasse
i. d. R.	in der Regel
MS	markt-studie.de
o. g.	oben genannt
OOP	Objekt Orientierte Programmierung
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RSS	Really Simple Syndication
SEM	Search Engine Marketing
SEO	Search Engine Optimazation
TF-IDF	Termfrequenz-Inverse Dokumentfrequenz
u. a.	unter anderem
v. a.	vor allem
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language
z. B.	zum Beispiel

1. Einleitung

1.1. Das Marktstudienportal www.markt-studie.de

Wie in dem Titel schon erwähnt, handelt es sich bei `www.markt-studie.de` um ein Portal bzw. Einkaufsportal für Marktstudien. Ein anderes signifikantes Synonym für Marktstudie ist der Begriff Marktforschung. In dieser Arbeit wird jedoch der Begriff Marktstudie bevorzugt und durchgängig benutzt.

Das Portal wird von dem Unternehmen dynamic Technologies GmbH (dytec GmbH) betrieben, das wiederum ein Tochterunternehmen der Luna Park GmbH ist. Die Luna Park GmbH[1] hat sich auf den Bereich des *Search Engine Optimization*[2] (SEO) und des *Search Engine Marketing*[3] (SEM) spezialisiert.

Die dytec GmbH verfasst die Marktstudien nicht selbst, die in dem Portal vorhandenen Daten erhält das Unternehmen von seinen Kunden bzw. Marktstudienanbietern, d. h. das Urheberrecht der einzelnen Marktstudien liegt immer bei den Marktstudienanbietern. In dem Portal werden nicht die kompletten Marktstudien veröffentlicht, sondern es werden lediglich der Titel, die allgemeine Beschreibung der Marktstudie, Inhaltsverzeichnis, Tabellenverzeichnis, Abbildungsverzeichnis, Preis und Format der Marktstudie (PDF, Druckversion) angezeigt. Folgender Link zeigt beispielhaft eine Studiendetailseite aus dem Portal:
`http://www.markt-studie.de/studien/immobilienfinanzierung-2009-p-3974.html`

Alles in allem bietet die dytec GmbH mit dem Portal markt-studie.de eine Dienstleistung für Marktstudienanbieter an, die ihre Marktstudien im Web vermarkten und verkaufen wollen. Die Bereitstellung und Veröffentlichung der Marktstudien in dem Portal ist kostenlos. Lediglich bei Verkauf einer Marktstudie erhält die dytec GmbH eine Provision.

1.1.1. Erläuterungen zum technischen Hintergrund

Das Marktstudienportal ist eine Webanwendung und basiert auf *PHP*/*MySQL*. Als Shop System verwendet man die in PHP geschriebene OpenSource eCommerce[4] Software *osCommerce*[5]. Die Speicherung der Marktstudiendaten erfolgt mit dem relationalen Datenbanksystem *MySQL*, das ebenfalls im Sinne von OpenSource frei zur Verfügung steht.

[1] `www.luna-park.de`
[2] Suchmaschinenoptimierung
[3] Suchmaschinenmarketing
[4] umgangsprachlich auch als Shopsystem bezeichnet
[5] `http://www.oscommerce.com/`

1.2. Ursprung des Portals als Projekt

Vor der Online-Veröffentlichung von markt-studie.de im Dezember 2002 war die Entwicklung des Portals zunächst ein Projekt zwischen den BBE Retail Experts[6] und der Luna Park GmbH. Die Unternehmensberatung BBE Retail Experts trat damals mit dem Auftrag an die Luna Park GmbH heran, ein Projekt zu entwickeln, mit dem man Marktstudien online vermarkten kann. Vorher war dies ein eigenständiges Projekt der BBE, das jedoch an unzureichenden[7] Marketingstrategien scheiterte.

Da die Luna Park GmbH Spezialisten im Bereich des SEO und des SEM beschäftigt und hier bereits einige Erfolge verbuchen konnte, erhofften sich die BBE Retail Experts durch die Zusammenarbeit, das Projekt der Marktstudienvermarktung doch noch erfolgreich gestalten zu können.

Das Vorbild des Projekts ist das Online-Versandhaus amazon.com, das u. a. mit dem Verkauf von Büchern große Erfolge erzielt. Das Ziel der Luna Park GmbH war dementsprechend, ein Online-Versandhaus speziell für Marktstudien zu entwickeln. Aus dieser Zielsetzung heraus entstand das Marktstudienportal markt-studie.de und im Zuge der Projektentwicklung wurde das Tochterunternehmen dytec GmbH gegründet, das sich einzig und allein auf den Betrieb des Marktstudienportals konzentriert. Zum Ende des Projekts hin trat die Luna Park gmbH das Portal komplett an die dytec GmbH ab, jedoch unterstützt die Luna Park GmbH die dytec GmbH weiterhin sowohl in technischen Bereichen als auch in den Bereichen SEO bzw. SEM. Im Jahre 2005 ging die dytec GmbH mit einer englischen[8] und einer spanischen[9] Version des Martkstudienportals online, um auch international Fuß zu fassen. Am Anfang belieferten die BBE Retail Experts die dytec GmbH auch mit aktuellen Marktstudien für diese englisch- und spanischsprachigen Pendants. Im Laufe der Jahre kamen andere Marktstudienanbieter hinzu und so stieg die Anzahl der Kunden bzw. Anbieter bei der markt-studie.de immer weiter an. Aktuell bietet das Portal Studien von 253[10] Studienanbietern an.

1.3. Die Marktstudien

An dieser Stelle ist anzumerken, dass Marktstudien branchenspezifisch sind, d. h. die Marktstudien sind einer oder auch mehreren Kategorien zugeordnet. Die Kategorien bei

[6] www.bbe-retail-experts.de
[7] s. [San10]
[8] www.reports-research.com
[9] www.estudio-mercado.es
[10] Quelle: MySQL Datenbank des Portals; Stand: 05.07.2010

markt-studie.de und ihren englisch- und spanischsprachigen Pendants orientieren sich an dem Mitbewerber `www.marketresearch.com`. Betrachtet man die Kategorien, dann stellt man fest, dass es sich hierbei um geläufige und bekannte Branchen handelt, wie z. B. *Industrie* oder *Konsumgüter*.

1.4. Import der Marktstudien

Für die Importe der Marktstudien in die Datenbank ist im Unternehmen die Redaktion verantwortlich. In unregelmäßigen Abständen senden die Kunden der Redaktion die Marktstudiendaten zu. Die Daten zu den Marktstudien werden zumeist in einer *Comma-seperated Values* (CSV) Datei hinterlegt. Nach dem Erhalt der Daten müssen die einzelnen Marktstudien in der CSV Datei manuell kategorisiert werden. Erst wenn dieser Arbeitsprozess abgeschlossen ist, wird ein Datenimport vorgenommen, der über den Administrationsbereich von *osCommerce* getätigt wird.

1.5. Neue Technologien für das Portal

Im Laufe der Jahre wurden auch neue Webtechnologien in das System des Studienportals implementiert: Mittlerweile bietet das Portal neben RSS Feeds für Marktstudien auch eine XML-Schnittstelle für den Datenaustausch mit einigen Kooperationspartnern an. Dank der XML Technologie sind die Studiendaten universell einsetztbar und austauschbar. Auf dieser Grundlage bieten sich weitere Projekte an, die zu einem späteren Zeitpunkt in dieser Diplomarbeit beschrieben werden.

Webtechnologien, die von dem Portal in jüngster Vergangenheit als flächendeckende und konstengünstige Marketingmaßnahmen[11] genutzt werden, sind die sogenannten *Web 2.0* Applikationen, hierzu gehören *Social Network* Seiten wie *Twitter*, *Facebook* und *Flickr*.

1.6. Der nächste Schritt für das Marktstudienportal

Nächstes großes Ziel von markt-studie.de ist der Vorstoß in das Semantic Web. Mit Hilfe der *Web Ontology Language* (OWL) Technologie können die Marktstudiendaten in einer Web Ontologie abgebildet werden. Jede Marktstudie wird auf dieser Grundlage ihrer Bedeutung zugeordnet. Die Web Ontologie zu den Marktstudien wird später für eine

[11] Virales Marketing

semantische Suchmaschine als Index bzw. Suchindex zur Verfügung gestellt. Das Unternehmen möchte auf Basis dieser Web Ontologie eine Wissensrepräsentation speziell für Marktstudien aufbauen.

1.7. Allgemeines zum Projekt

Der in Kapitel 1.4 beschriebene Arbeitsprozess zur manuellen Kategorisierung stellt sich als sehr komplex und zeitintensiv dar. Die für die manuelle Kategorisierung benutzte CSV Datei beinhaltet i. d. R. mehrere tausend Daten. Dabei ist zusätzlich zu bedenken, dass derartige Daten zeitnah von mehreren Kunden zugesendet werden. Zur Optimierung des Kategorisierungsvorgangs soll eine Java Anwendung entwickelt werden, die die Redaktion bei der Kategorisierung unterstützt. Die Anwendung soll anhand des jeweiligen Marktstudientitels und der zugehörigen Beschreibung die Kategorisierung vornehmen. Zusätzlich dazu sollen im Rahmen der Erweiterung des Marktstudienportals in das Semantic Web die Marktstudiendaten mitsamt ihrer Kategorienzuordnung als Web Ontologie gespeichert werden. Genauer: Aus den Daten soll eine OWL Datei generiert und abgespeichert werden. Im Großen und Ganzen soll die Kategorisierung semi-automatisch von statten gehen, d. h. die Anwendung unterbreitet dem Benutzer einige Vorschläge, anhand derer er die Marktstudien einer Kategorie oder mehreren Kategorien zuordnen kann. Die zugehörigen Kategorien sind schon vorhanden und müssen nicht mehr neu definiert werden.

Bei der Kategorisierung kommt die *Text Extraction* zum Einsatz, mit der man aus einem Text die signifikanten Wörter ermittelt und anhand derer wiederum die Zuordnung der Marktstudien in eine Kategorie oder mehrere Kategorien erfolgt. Die Frage, ob diese ermittelten signifikanten Wörter zu einer Kategorie passen, ist Kern dieses Projektes. Dabei werden später unterschiedliche Lösungsvorschläge und -techniken aus der *Text Extraction* und ggf. aus dem *Text Mining* vorgestellt, mit denen man bestimme Begriffe exakt einer Kategorie zuordnet. Weiterhin werden die vom W3C standardisierten Techniken des Semantic Web wie z. B. OWL verwendet.

Da man zuerst eine Grund Ontologie aufbauen muss, bevor man in die Text Analyse geht, erfolgt zunächst eine Darstellung des Themas Semantic Web.

Mit diesem Projekt soll keineswegs das bestehende Marktstudienportal durch das Semantic Web ersetzt werden. Es wird lediglich eine Erweiterung des Portals vorgenommen, mit dem Ziel, die Studiendaten auch als eine Web Ontologie anzubieten und die Kategorisierung zu erleichtern.

2. Semantic Web

In diesem Kapitel werden die Hintergründe und Ursprünge des Semantic Web und des herkömmlichen Web sowie deren Problematik beleuchtet. Weiterhin werden die grundlegende Ideen des Semantic Web beschrieben und seine zentralen Begriffe aufgegriffen und erläutert.

2.1. Hintergrund: Die Bedeutung des Web in der heutigen Zeit

Kaum eine andere Technologie hat das Alltagsleben und den Umgang mit Informationen so schnell und gravierend verändert wie das *World Wide Web* (WWW). Durch die rasante Entwicklung des Web wandelte sich die Industriegesellschaft allmählich zu einer Informationsgesellschaft[12]. Das Web veränderte auch die Art und Weise, wie die Menschen miteinander kommunizieren, wie Informationen verbreitet und wiedergefunden werden und es beeinflusste ebenfalls die Ausführung von Geschäften bzw. Geschäftsbeziehungen in Unternehmen[13].

Dank einer Standardisierung des Web[14] und der darauf aufbauenden Infrastruktur zeichnet sich das Web durch eine Reihe von erheblichen Vorteile aus. Folgende Merkmale sind für das Web prägend: „Aktualität und Verfügbarkeit“[15]. Unter Aktualität versteht man, dass Informationen zu jeder Zeit auf den neuesten Stand gebracht werden und durch die Verfügbarkeit ist u. a. die Lokalität von Informationen bzw. Informationsquellen kaum von Belang, da man auf eine beliebige bereitgestellte Information irgendwo im Web von jedem Ort aus zugreifen kann. Diese Tatsache wird auch als universelle Verfügbarkeit bezeichnet[16]. Dadurch stellt sich der Weg zur Informationsbeschaffung weniger aufwändig und kostengünstiger dar und darüber hinaus sind die Informationen für eine breitere Bevölkerungsschicht öffentlich zugänglich.

Die zunehmende Bedeutung des Web sowohl in kommerzieller als auch in gesellschaftlicher Hinsicht führte zu einer Liberalisierung der Bereitstellung von Informationen. Vor dem Zeitalter des WWW gestaltete sich die Verbreitung und Veröffentlichung von Informationen für das breite Publikum als ausgesprochen schwierig, da Informationen nur einigen bestimmten Gruppen, genauer gesagt den sogenannten „Informationsoligopolisten“[17],

[12] [HKRS08], S. 15
[13] [AH08], S. 20
[14] `http://www.w3.org/standards/webarch/`
[15] [HKRS08], S. 9
[16] ebenda, S. 9
[17] ebd., S. 9

vorbehalten waren. Heutzutage werden solche Schranken durch das Web umgangen – es bietet einzelnen Personen oder auch kleineren Interessengruppen die Möglichkeit, ihre Informationen dem breiten Publikum zur Verfügung zu stellen, ohne irgendwelche Kosten dafür aufbringen zu müssen.

All Diese Vorzüge und Vorteile des Web spiegeln die Ideen des WWW Erfinders Tim Berners-Lee wider. In seinem „WorldWideWeb: Summary“[18] beschreibt Tim Berners-Lee das Web folgendermaßen: „The WWW project merges the techniques of information retrieval and hypertext to make an easy but powerful global information system“[19]. Dies bedeutet: Das WWW Projekt wird mit den Techniken des Information Retrieval und den Ideen des Hypertext zu einem globalen Informationssystem vereinigt bzw. vermischt. Die Betonung liegt hier auf dem Begriff „Global Information System“, also ein System, in dem Informationen gespeichert, wiedergefunden und einer breiten Masse zur Verfügung gestellt werden. Ein anderer Pionier des WWW, Kevin Hughes, beschreibt das Web in seinem Paper[20] noch vortrefflicher: „The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents“[21]. Kevin Hughes interpretiert das WWW als eine Art universelles Dokumentenarchiv, auf das immer und zu jeder Zeit zugegriffen werden kann. D. h. jeder Benutzer eines Computernetzwerks ist in der Lage, ständig auf eine Reihe von Medien zurückgreifen zu können.

Heutzutage ist die Benutzung des WWW kaum aus dem Alltagsleben wegzudenken. Aktuell existieren diverse kommerzielle Nutzungsmöglichkeiten des Web wie z. B. das Einkaufsportal `www.amazon.de` oder auch unabhängige Nachrichtenportale. In letzter Zeit gewann das sogenannte Social Networking durch Portale wie Facebook oder Twitter an Bedeutung.

Das Marktstudienportal markt-studie.de ist ein typisches Beispiel für ein Einkaufsportal, also eine kommerzielle Nutzung des Web. Auch das Portal nutzt die Annehmlichkeiten und Vorteile des Web. Die Veröffentlichung der Studien im Studienportal ist mit keinen Kosten verbunden und die Daten werden nach einem bestimmten Monatsrhythmus ständig aktualisiert. Mit Hilfe der Social Networking Portale können die Links zu den Studien verbreitet werden, was dazu führt, dass diese Daten eine stärkere Präsenz im Web erhalten. Dies spiegelt die beiden o. g. prägnanten Merkmale des Web wider: „Aktualität“ und „Verfügbarkeit“.

[18] `http://groups.google.com/group/alt.hypertext/msg/395f282a67a1916c`
[19] ebenda
[20] s. [HUG95]
[21] ebenda

2.2. Nachteile des herkömmlichen Web

Trotz aller Vorzüge und Vorteile des Web tauchen doch auch etliche Probleme auf. Man darf die Tatsache nicht ignorieren, dass die im Web vorhandenen Informationen nur von Menschen konsumiert werden und die Lesbarkeit auch nur auf den Menschen beschränkt ist[22]. Ein gewöhnlicher Web-Content beispielsweise, der aus einer Datenbank erzeugt wird, stellt den Inhalt ohne strukturelle Informationen dar. Maschinen können mit den unstrukturierten Ansammlungen von Daten im Web nichts anfangen, sie können sie nicht weiterverarbeiten. Ein Mensch als Endkonsument der Informationen besitzt die Fähigkeit, auf einer Webseite gefundene Informationen zu interpretieren und sie mit anderen Informationen in Beziehung zu setzen, während auf der anderen Seite die Maschine ohne strukturelle Grundlage der Informationen nicht in der Lage ist, diese Aufgaben zu erfüllen[23]. So kann ein Mensch im Gegensatz zu der Maschine auf Basis einer Information Schlussfolgerungen ziehen und diese zu einem Gesamtergebnis zusammenfassen.

Das Prinzip der Verfügbarkeit, im letzten Kapitel explizit erläutert, bedarf ebenfalls einer näheren Betrachtung. Die Informationen im Web sind zwar prinzipiell überall verfügbar und dem Zugriff darauf sind keine Grenzen gesetzt, dennoch ist es problematisch und schwer, die explizit gesuchten Informationen in der Menge der Informationen, die das Web hergibt, zu finden. Dieses Problem bezieht sich auf Suchmaschinen wie Google oder Yahoo. Das Suchen und Finden von Informationen zu Recherchezwecken ist heutzutage eine typische Webnutzung. Hinter der Suchmaschine verbirgt sich ein ausgefeilter Algorithmus zum Auffinden von Information, letztendlich aber basiert die Suchmethode auf dem Lokalisieren von Zeichenketten im Text[24], es handelt sich also bei den gängigen Suchmaschinen um eine stichwortbasierte Suche. Mit der stichwortbasierten Suche erzielt man zwar eine hohe Trefferquote, jedoch sind die Ergebnisse wenig präzise. Aus Erfahrung ist diese Feststellung äußerst zutreffend, da i. d. R. nach einer Suchanfrage bei Google zunächst mehr als Tausende Ergebnisse aufgelistet werden – die meisten dieser Informationen sind nicht relevant und somit kaum nutzbar[25]. Manchmal kommt es sogar vor, dass überhaupt keine Ergebnisse geliefert werden, so z. B wenn das gesuchte Schlüsselwort in den durchsuchten Web-Dokumenten nicht auftaucht. Diese Feststellung zeigt wie stark vokabularabhängig[26] die Suchergebnisse in der stichwortbasierten Suche sind. Eine Alternative bietet die inhaltsbezogene, also semantische Suche. Beispielswei-

[22] s. [BER01]
[23] [HKRS08], S. 12
[24] ebenda, S.10
[25] Bei der Suche nach dem Begriff „Kohl“ in Google erhält man als Ergebnis sowohl das Gemüse als auch den Alt-Bundeskanzler
[26] [AH08], S. 2

se sollten die Suchergebnisse für den Suchbegriff „Wein“ nicht nur alle Informationen zu Wein liefern, sondern es sollten auch andere alkoholische Getränke oder Weinarten in den Ergebnissen angezeigt werden.

Oftmals kommt es aber vor, dass eine exakte Information, nach der ein Nutzer sucht, nicht explizit auf einer einzelne Webseite oder im Web zu finden ist, sondern die kompletten Informationen aus unterschiedlichen, über das Web verteilten Fakten bestehen. Der Nutzer müsste dann die einzelnen Seiten durchsuchen, um die passenden Fakten zusammenzutragen, aus denen er letztlich Schlüsse ziehen kann. Diese Problematik bezeichnet man als Problem des impliziten Wissens. Implizites Wissen wird auch als stilles Wissen bezeichnet und leitet sich aus dem englischen „tacit knowledge“ ab. Nach [RAN01] definiert sich implizites Wissen folgendermaßen: „Es sind Kenntnisse oder Fähigkeiten, die nicht explizit formuliert sind“[27]. Das Wissen lässt sich nur *anzeigen* und ist auch nicht *erklärbar* oder es ist kaum möglich es in Worten auszudrücken[28]. Beispielsweise findet ein Nutzer erst am Ende seiner Faktensammlung unerwartet eine zusätzliche Information, die sich jedoch als eine gute Ergänzung zur eigentlichen Zielinformation herausstellt.

Nicht ausser acht zu lassen ist die Problematik der verschiedenen Kodierungstechniken, Dateiformate, natürlichen Sprachen und der unterschiedliche Aufbau privater Homepages – der unzähligen Webseiten im Web. Die Gründe dafür sind die dezentrale Struktur und Organisation des Web, dies hat zur Folge, dass die vorhandenen Informationen im Web heterogen sind. Dies macht es schwierig, über das Web verteilte Informationen zu sammeln, sie zusammenzufassen und ggf. weiterzuverarbeiten. Als Anwendungsbeispiel könnte man hier den Versuch, die verschiedenen politischen Programme der deutschen Parteien auf Basis ihrer Internetpräsenz miteinander zu vergleichen, nennen. Dieses Vorhaben wird schwer zu bewerkstelligen sein, da eventuell die Webseiten der verschiedenen Parteien unterschiedlich aufgebaut sind. Verfolgt man dieses Beispiel weiter und möchte die Parteiprofile der deutschen und der isländischen konservativen Partei miteinander vergleichen, scheitert dies schon an den unterschiedlichen natürlichen Sprachen, da Island ein anderes Alphabet benutzt. Dies bezeichnet man als das Problem der „Informationsintegration“[29].

Um diesen verschiedenen Problematiken entgegenzuwirken, entstand beim W3C das Semantic Web Projekt, das auf dem Vorschlag des WWW Erfinders Tim Berners-Lee beruht.

[27] [RAN01], S. 1
[28] ebenda, S. 2
[29] [HKRS08], S. 10

2.3. Semantic Web Initiative

Das Konzept und der Vorschlag zum Semantic Web beruhen auf einem Artikel von Tim Berners-Lee, den er gemeinsam mit James Hendler und Ora Lassila verfasst und im amerikanischen Wissenschaftsmagazin „Scientific American“[30] veröffentlicht hat.

In dem Artikel spricht Tim Berners-Lee die verschiedenen Problematiken des herkömmlichen Web an. Zum einen die Feststellung, dass Informationen im Web einschließlich und nur für den Menschen bestimmt sind und zum anderen, dass Maschinen nicht in der Lage sind, diese Informationen weiterzuverarbeiteten.

Die Hauptidee des Semantic Web besteht darin, das homogen verteilte Web aus der Ebene der Darstellung in die Ebene der Daten zu heben.

Das Semantic Web soll jedoch das herkömmliche Web nicht ersetzen oder separat zum bestehenden Web existieren, es stellt lediglich es eine Erweiterung des herkömmlichen Web dar, in der Informationen Bedeutungen zugewiesen werden, so dass Maschinen die Informationen „verstehen“ und weiterverarbeitet können.

In seinem Artikel erläutert Tim Berners-Lee das Semantic Web am Beispiel von zwei Geschwistern, die einen passenden Arzt für ihre Mutter finden wollen. Wenn geeignete Ärzte gefunden werden, dürfen die Arzttermine sich nicht mit den Terminen der Geschwister überschneiden. In seinem Beispiel hebt Tim Berners-Lee besonders die Rolle des sogenannten Software Agenten hervor. Dieser Agent wird von beiden Geschwistern beauftragt, im Semantic Web die relevanten Informationen zu finden. Laut Tim Berners-Lee nimmt der Software Agent eine tragende Rolle im Semantic Web ein. Die Potenz des Semantic Web wird erst entfaltet, wenn viele Entwickler sich auf die Entwicklung eines Programms konzentrieren, die einzig und allein die Aufgabe hat, Web Inhalte aus verschiedenen Quellen zu sammeln, weiterzuverarbeiten und die Informationen mit anderen Programmen auszutauschen. Je mehr solcher Programme im Semantic Web kursieren, desto effektiver arbeitet ein Software Agent[31].

Die beiden Autoren Grigoris Antoniou and Frank van Harmelen griffen in ihrem Buch „A Semantic Web Primer“[32] das Beispiel von Tim Berners-Lee auf und erläuterten die Problematik der Arztsuche in der praktischen Umsetzung.

Eine gewöhnliche Webseite wird mittels der Auszeichnungssprache[33] HTML aufgebaut und strukturiert. Solche Hypertext Dokumente werden dann von einem Webbrowser in-

[30] s. [BER01]
[31] [BER01], S. 4
[32] s. [AH08]
[33] engl.: Markup Language

terpretiert und dargestellt. Um noch einmal auf das Arztbeispiel zurückzugreifen, zeigt folgender HTML Code grob und beispielhaft die Einstiegsseite einer Arztpraxis:

```
<h1>Agilitas Physiotherapy Centre</h1>
Welcome to the Agilitas Physiotherapy Centre home page.
Do you feel pain? Have you had an injury? Let our staff
Lisa Davenport, Kelly Townsend (our lovely secretary)
and Steve Matthews take care of your body and soul.
<h2>Consultation hours</h2>
Mon 11am - 7pm<br>
Tue 11am - 7pm<br>
Wed 3pm - 7pm<br>
Thu 11am - 7pm<br>
Fri 11am - 3pm<p>
But note that we do not offer consultation
during the weeks of the
<a href=". . .">State Of Origin</a> games.
```

Listing 1: HTML-Code zu einer Einstiegswebseite

Für Menschen ist diese Information lesbar und konsumierbar. Der Inhalt besteht aus einem Haupttitel, einem Text, einem Untertitel und Informationen zu den möglichen Terminen. Maschinen wären nicht fähig oder nicht zuverlässig genug, diese Angaben zu interpretieren und zu verarbeiten. Die stichwortbasierte Suche kann zwar die Schlüsselwörter wie „Physiotherapeut“ oder „Sprechzeiten“ herausfiltern, jedoch würde die stichwortbasierte Suche den Unterschied zwischen der Sekretärin und dem Therapeuten nicht erkennen.

Im Semantic Web werden diese Informationen zunächst in einer speziellen Struktur zur Verfügung gestellt, mit der die Maschinen auf verlässliche Weise arbeiten können.

Wandelt man das vorherige Beispiel in eine spezielle Struktur um, könnte die Lösung wie folgt aussehen:

```
<treatmentOffered>Physiotherapy</treatmentOffered>
<companyName>Agilitas Physiotherapy Centre</companyName>
<staff>
<therapist>Lisa Davenport</therapist>
<therapist>Steve Matthews</therapist>
<secretary>Kelly Townsend</secretary>
</staff>
</company>
```

Listing 2: XML Gegenstück zum HTML Code in Listing 1

Wie aus dem Beispiel zu entnehmen ist, sind wichtige Informationen wie der Name des Therapeuten oder der Sekretärin mit Hilfe von XML ausgezeichnet, damit die Maschinen den Unterschied erkennen können. Solche Daten bezeichnet man auch als Metadaten, also Daten, die Aussagen über andere Daten machen. Diese Struktur ist für Maschinen erkennbar, sie können damit auf zuverlässiger Art und Weise weiterarbeiten. In Kapitel 3.1 wird näher auf XML eingegangen.

Da im Web verschiedene Webseiten ihre Informationen unterschiedlich aufbereiten, müssen einheitliche und offene Standards definiert werden, um eine Interoperabilität zwischen den verschiedenen Plattformen und Anwendungen zu erreichen. D. h. es muss eine Möglichkeit gewährleistet werden, die Informationen aus verschiedenen Quellen über Anwendungs- und Datenformatgrenzen hinweg zusammenzutragen, auszutauschen und ggf. miteinander in Verbindung zu setzen.

Um genau solche Standards für das Web zu definieren, rief das W3C das Semantic Web Projekt ins Leben. Aus dem Projekt entstanden die grundlegenden Standards für Informationsspezifikationen wie XML, RDF(S)[34] und OWL. Die XML Technologie ist zwar nicht dem Semantic Web sondern eher dem herkömmlichen Web zuzuordnen, jedoch bietet XML eine syntaktische Grundlage für das Semantic Web. Im Gegensatz dazu sind RDF(S) und OWL sogenannte Ontologie-Sprachen, die speziell für das Semantic Web entwickelt wurden. Dem Begriff Ontologie begegnet man sehr oft im Semantic Web. In Zusammenhang mit dem Semantic Web bedeutet Ontologie Wissensbasis, die nichts anderes ist als ein in RDF(S) und OWL erstelltes Dokument, welches Wissen repräsentiert. In Kapitel 2.4 wird auf den Begriff Ontologie eingegangen.

Ein weiteres Merkmal im Semantic Web ist die automatische Schlussfolgerung neuer Fakten aus vorangegangenen Informationen. Die englische Bezeichnung lautet „Reasoner“ – es geht darum, implizites Wissen in explizites Wissen umzuwandeln[35]. Dies behebt das Problem des impliziten Wissens, das man sehr oft in der Anwendung des herkömmlichen Web findet. Diese Thematik stammt aus dem Bereich der formalen Logik.

Mit der Idee des Semantic Web traten auch einige Missverständnisse auf. Es gibt zahlreiche Äusserungen[36], die behaupten, durch das Semantic Web können Maschinen bzw. Computer tatsächlich in die Lage versetzt werden, die Bedeutung von Informationen zu verstehen. Die Grundidee des Semantic Web verfolgt jedoch eine moderate und einfache Zielsetzung: Mit Hilfe des Semantic Web sollen die Probleme des herkömmlichen Web behoben werden, ohne das vorhandene Web zu ersetzen oder eine parallele Technologie

[34] s. S. 30 K. 3.1.1
[35] [AH08], S. 13
[36] [HKRS08], S. 12

neben dem klassischen Web zu etablieren. Der Grundgedanke besteht darin, Wege und Methoden zu finden, Informationenen in eine spezielle Form zu bringen und zu repräsentieren, mit der Maschinen auf eine zuverlässige Art und Weise umgehen können. Im Semantic Web geht es nicht nur um das Verstehen der Informationen seitens der Maschinen, die Maschinen sollen vielmehr die Informationen so aufbereiten, dass die Menschen einen Nutzen daraus ziehen können. Ziel ist eine bessere Kommunikation bzw. Interaktion zwischen Mensch und Maschine. Aus der Sicht von Tim Berners-Lee kann mit der Unterstützung von Ontologien das Arbeiten im Web verbessert und aufgewertet werden. Ontologien können benutzt werden, um die Genauigkeit und Treffsicherheit bei einer Websuche zu erhöhen.

2.4. Ontologien

Im Semantic Web kommt man nicht um den Begriff „Ontologie“ herum. Die Ontologie ist die Grundlage für die von Tim Berners-Lee vorgeschlagene Idee des Semantic Web. Die Herkunft des Begriffes stammt aus der Philosophie – schlägt man dieses Wort in einem Fremdwörterlexikon nach, so stößt man auf folgende Definition: „Lehre vom Sein, vom Wesen und den Eigenschaften des Seienden“[37]. Im Großen und Ganzen beschäftigt sich Ontologie mit der Identifizierung, Beschreibung und Benennung von Dingen bzw. mit der Existenz von Dingen, die in der Welt vorhanden sind.

Der Begriff Ontologie tauchte in den letzten Jahren auch in der Informationstechnologie auf. V. a. im Bereich der Wissensrepräsentation[38] ist die Ontologie eine wichtige Komponente, um Wissen zu modellieren. In der Informatik jedoch wird die Definition von Ontologie ein wenig abgewandelt. Der amerikanische Informatiker und Pionier im Bereich der Wissensrepräsentation, Tom Gruber definiert Ontologie in der Informatik folgendermaßen: „An ontology is an explicit specification of a conceptualization“[39]. Allgemein betrachtet bezieht sich eine Ontologie auf eine genau definierte Domäne von Gegenstandsbereichen[40]. I. d. R. besteht eine Ontologie aus einer Reihe endlicher Listen von Begriffen bzw. Klassen und deren Beziehungen zueinander. Eine wichtige Rolle bei Ontologien spielen die sogenannten Konzepte, die Klassen von Objekten in einer Domäne repräsentieren. Folgendes Beispiel zeigt vereinfacht eine Ontologie zu dem Begriff „Getränke“, der aus der Marktstudienkategorie entnommen ist.

[37] [Lec08], S. 251
[38] engl.: Knowledge Representation
[39] s. [GRU92]
[40] engl.: Universe Of Discourse

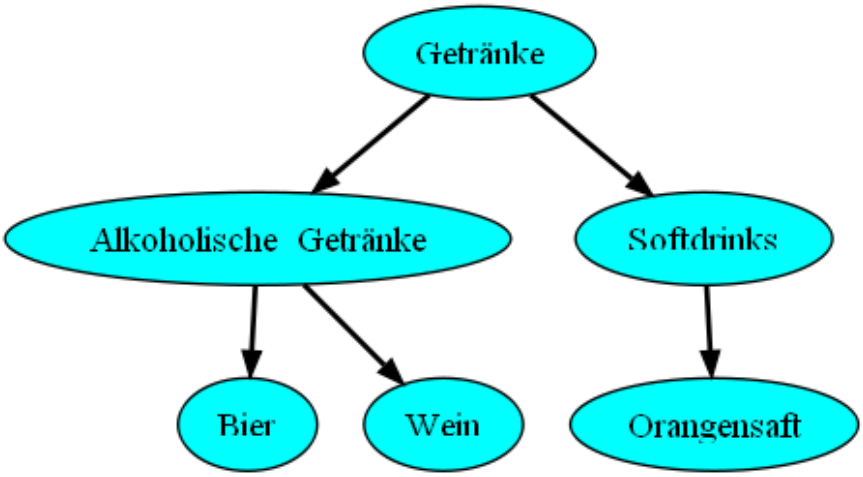


Abbildung 2.1: Die Ontologie zu dem Begriff „Getränke“

Die Beziehungen zwischen den Begriffen bzw. Klassen und Unterklassen in einer Ontologie werden meistens hierarchisch aufgebaut.

In Bezug auf das Semantic Web können mittels einer Ontologie gemeinsame Begriffsvereinbarungen zu einer Domäne getroffen werden. Die Autoren des Buches „A Semantic Web Primer“ bezeichnen dies als „Shared Understandings“[41], also ein Weg zum gemeinsamen Verständnis von Gegenständen und Objekten eines Wissensbereichs. Beispielsweise benutzt eine Applikation den Begriff „Marktstudie“, während eine andere Applikation den Begriff „Marktforschung“ verwendet. Mit Hilfe der Ontologie kann man vereinbaren, dass sowohl „Marktstudie“ als auch „Marktforschung“ Unterbegriffe von „Studien“ sind, so dass die beiden Applikationen problemlos miteinander kommunizieren und die Daten austauschen können, sofern sie sich auf die gleiche Domäne beziehen. Dies führt zu einer Interoperabilität zwischen den verschiedenen Applikationen. Ein weiteres Problem stellen Homonyme dar: Z. B. verwendet eine Webseite den Begriff „Number“ und bezieht sich damit auf eine Hausnummer, eine andere Webseite ordnet dem Begriff „Number“ eine Postleitzahl zu. Die Interoperabilität zwischen den beiden Webseiten ist hier nicht mehr möglich, doch dieses Problem kann ebenfalls durch Ontologien gelöst werden, indem man die Begriffe genauer spezifiziert, also statt „Number“ dann „House-Number“ bzw. „Postcode“ verwendet.

Neben den Beziehungen zwischen Klassen und Unterklassen beinhaltet eine Ontologie folgende weitere Informationen:

- Eigenschaften
- Einschränkungen von Werten

[41] [AH08], S. 12

- disjunkte Aussagen
- Festlegung von logischen Beziehungen zwischen Objekten

Bei näherer Betrachtung des Themas entdeckt man eine gewisse Ähnlichkeit mit der Objekt Orientierten Programmierung (OOP). Wie bei Ontologien entwickelt man in der OOP ebenfalls Konzepte zu Klassen, die eine Art Blaupause für Objekte darstellen. In der OOP besteht auch die Möglichkeit, zwischen den Objekten bzw. zwischen den Eigenschaften von Objekten diverse Beziehungen zu definieren. Der Unterschied zwischen den beiden liegt in der Herkunft: Während die OOP eher ein Programmierparadigma beschreibt, stammt die Ontologie aus der Philosophie, genauer gesagt ist sie ein Bestandteil der theoretischen Philosophie, die sich mit der Frage des Seins beschäftigt. Mit Unterstützung der Ontologien erhofften sich viele Fachleute aus dem Bereich der künstlichen Intelligenz, existenzielle Dinge aus der Welt oder auch die natürlichen Sprachen in Computern abzubilden.

3. W3C Standards

In den nächsten Kapiteln werden die W3C Standards für Web Ontologien beschrieben, es werden wichtige Merkmale und Definitionen aufgegriffen und erläutert.

3.1. XML

XML ist eine Abkürzung und steht für *Extensible Markup Language*, was übersetzt soviel wie „erweiterbare Markierungssprache“ heißt.

Die XML Technologie wurde von dem W3C mit dem Ziel, die Grenzen der verschiedenen proprietären Datenformate zu sprengen[42], entwickelt und standardisiert. Vor dem XML Standard speicherten Programme zur Textverarbeitung oder Tabellenkalkulation ihre Daten in ihren eigenen proprietären Datenformaten ab, die von anderen Programmen nicht gelesen oder kaum bearbeitet werden konnten, so dass ein Datenaustausch[43] zwischen den Anwendungen kaum möglich war. Zwar gibt es im Internet oder auch im Intranet eines Unternehmens Wege, wie man Daten unterschiedlicher Formate zwischen den Anwendungen austauschen und verteilen kann, dieses Unterfangen ist jedoch ohne eine ergänzende Software kaum oder gar nicht möglich.

Das W3C[44] definiert durch die XML Standards eine Empfehlung[45] [46], um eine bestimmte Art von maschinenlesbaren Dokumenten zu entwickeln. Die Recommendation beinhaltet Spezifikationen zum Aufbau eines XML Dokuments und beschreibt außerdem wie andere Software sich bei der Bearbeitung von XML Dokumenten verhalten soll.

Bei XML handelt es sich um eine Auszeichnungssprache, die dazu dient, die in einem Textdokument befindlichen Daten zusätzlich mit Informationen zu versehen. Diesen Vorgang nennt man auch „auszeichnen“ oder „annotieren“[47]. Solche mit Zusatzinformationen versehenen Daten bezeichnet man als Metadaten, da es sich um Daten handelt, die andere Daten beschreiben[48].

Eine sehr geläufige und bekannte Auszeichnugssprache, mit der man ein Hypertext Dokument erstellen kann, ist HTML. Der Schwerpunkt dieser Sprache liegt in der Spezifikation von Informationen zur Darstellung von Dokumenten. In einem Hypertext Dokument

[42] [Erl01], S. 12
[43] Interoperabilität
[44] `www.w3c.org`
[45] engl.: Recommendation
[46] `http://www.w3.org/TR/REC-xml/`
[47] engl.: annotation
[48] vgl. S. 21 K. 2.3

werden die zugehörigen Textkomponenten mit Hilfe von sogenannten „Tags“ ausgezeichnet. Folgendes HTML Beispiel ist dem Kapitel 2.3 entnommen:

```
1 <h1>Agilitas Physiotherapy Centre</h1>
2 <b>Welcome to the Agilitas Physiotherapy Centre home page.</b>
```

Listing 3: HTML Code zu einer Einstiegswebseite aus Kapitel 2.3 entnommen

Ein Webbrowser würde dieses Beispiel wie folgt darstellen

Abbildung 3.1: Die Ausgabe des HTML Codes im Web-Browser

Die von der W3C verabschiedeten HTML Spezifikationen[49] beinhalten eine Reihe von Beschreibungen zu bestimmten Mengen solcher Tags und den dazugehörigen Eigenschaften[50]. Auf Basis dieser Definitionen werden dann Softwareprogramme wie Web-Browser programmiert, die diese Tags interpretieren und die in den Tags definierten Formatierungen darstellen. Aus der HTML Spezifikation geht hervor, dass die Menge der Tags festgelegt ist, also aus einem festen Vokabular zur Strukturierung von Texten im Hypertext Dokument besteht. Durch die ausgezeichneten Texte in HTML werden den Programmen wie Web-Browsern Hinweise gegeben, wie die annotierten Texte dargestellt werden sollen. HTML dient einzig und allein dazu, das Erscheinungsbild von Hypertext Dokumenten im Web-Browser zu beeinflussen.

Im Gegensatz zu HTML besitzt XML keine fest definierte Menge an Tags, sondern erlaubt allgemein das Definieren von Auszeichnungssprachen. Wie HTML verwendet auch XML Tags zur Textauszeichnung, dabei sind die Bezeichnungen von Tags je nach Anwendungsbereichen frei wählbar und aus diesem Grund existiert bei XML kein festes Vokabular. Der Schwerpunkt von XML liegt nicht in der Darstellung von Dokumenten, sondern in der Definition der logischen Struktur eines Dokuments. Laut der XML Essentials[51] vom W3C wird XML am besten zur Strukturierung und Repräsentation von Informationen wie Daten, Dokumenten, Konfigurationen, Büchern etc. verwendet. Die logische Struktur von XHTML Dokumenten wird durch XML definiert und in der Tat ist XHTML eine auf XML basierende Version von HTML.

[49] http://www.w3.org/TR/html4/
[50] engl.:Attributes
[51] http://www.w3.org/standards/xml/core

Die Effizienz von XML entfaltet sich beim Austausch und der gemeinsamen Verteilung von strukturierten Informationen im Web. Im XML Essentials des W3C wird dies als „shared structured information“[52] bezeichnet. Der Austausch von Informationen findet zwischen Computern, Menschen und zwischen Mensch und Computer statt. Die Kommunikation zum Informationsaustausch kann sich sowohl lokal als auch über Netzwerke ereignen. Folgendes Beispiel ist dem Kapitel 2.3 entnommen:

```
<treatmentOffered>Physiotherapy</treatmentOffered>
<companyName>Agilitas Physiotherapy Centre</companyName>
<staff>
<therapist>Lisa Davenport</therapist>
<therapist>Steve Matthews</therapist>
<secretary>Kelly Townsend</secretary>
</staff>
</company>
```

Listing 4: XML Code aus Listing 2

In diesem Musterbeispiel werden frei definierte Tags wie „<companyName>“ oder „<therapist>“ anstelle der HTML Tags verwendet. Diese Daten dienen als Metadaten für die Webseite einer Arztpraxis bzw. Therapeutenpraxis. Im Sinne von „shared structured information“ kann man dieses XML Dokument zum Austausch mit anderen Ressourcen wie Webseiten eines anderen Therapeuten anbieten. Betrachtet man erneut das Arztbeispiel von Tim Berners-Lee, so benutzt man diese Metadaten, um Suchagenten, die speziell für die Arztsuche konzipiert sind, besser zu unterstützen. Die ausgezeichneten Texte im XML Beispiel weisen den Suchagenten darauf hin, welche Behandlung dieser Arzt anbietet (Tag „<treatmentOffered>“) und wer der Therapeut ist (Tag „<therapist>“).

Allgemeinen betrachtet bietet XML die Möglichkeit, auf eine einfache und universelle Art Daten zu speichern. Die in XML hinterlegten Daten können über Anwendungsgrenzen hinweg elektronisch verarbeitet und verbreitet werden, womit sich XML als universelles Datenaustauschformat anbietet.

XML kann in unterschiedlichen Bereichen eingesetzt werden. Der Einsatz von XML reicht von anwendungsspezifischen XML Vokabularen, z. B. Programmkonfigurationen oder elektronisches Bestellformular, bis hin zu bereichsspezifischen XML Vokabularen, wie z. B. Datenaustauschformate für Naturwissenschaftler.

Der Vollständigkeit halber und aus historischen Gründen sei noch auf die Metasprache SGML hingewiesen, die schon seit über 20 Jahren existiert. Im Gegensatz zu XML ist die SGML Sprache komplexer und umfangreicher, da etliche selten gebrauchte Sprachmittel

[52] http://www.w3.org/standards/xml/core

verwendet werden. Viele Unternehmen zierten sich davor, eine Software zu bauen, die SGML Dokumente interpretiert, da durch die Unhandlichkeit und den Umfang, die SGML mit sich bringt, die Entwicklung von solchen Softwareprogrammen umfangreich und kostenintensiv gewesen wäre[53]. Die Entwickler vom W3C griffen stattdessen auf eine Teilspezifikation von SGML zurück und schufen daraus XML, die eine Art „leichtgewichtige“[54] Version von SGML darstellt. Folgende Grafik zeigt die verwandtschaftlichen Beziehungen zwischen den verschiedenen Auszeichnungssprachen.

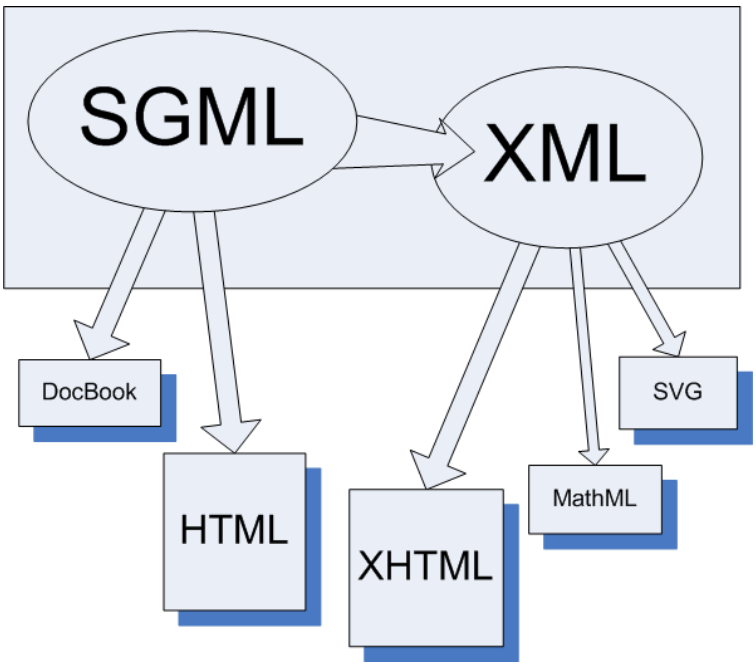


Abbildung 3.2: Die Beziehungen zwischen den verschiedenen Auszeichnungssprachen

Neben XML baut auch das klassische HTML auf SGML auf. Auf den Grundlagen von XML entwickelte man XHTML, also eine auf XML basierende HTML Version. Die Auszeichnungssprache *MathML* ist ein gutes Beispiel für ein bereichsspezifisches XML Vokabular und wird verwendet, um mathematische Formeln zwischen Naturwissenschaftlern auszutauschen.

3.1.1. Reicht XML als Ontologie-Sprache aus ?

Die XML Technologie ist eine standardisierte und weit verbreitete Metasprache, die zur Erzeugung maschinenlesbarer Dokumente dient und eine gute Grundlage für den Aufbau eines strukturierten Dokuments bietet. Die größte Stärke von XML, ihre Verwendung als

[53] [HKRS08], S. 19
[54] [Erl01], S. 12

universelles Datenaustauschformat, ist auch ihre größte Schwäche. Die freie Auswahl von Tag-Namen bietet zwar eine gute Möglichkeit, Textelemente eines XML Dokuments auszuzeichnen, bietet allerdings aus Sicht des Semantic Web keine Vorteile gegenüber der natürlichen Sprache. Die Bezeichnungen von Tags sind näher betrachtet nichts anderes als Wörter, deren Bedeutungen auch mehrdeutig sein können und deren Beziehungen zueinander nicht eindeutig festgelegt werden können.

Aus menschlicher Perspektive sind die Bedeutungen der einzelnen Tag-Bezeichnungen zwar verständlich, aus Sicht der Maschinen aber sind die Bezeichnungen im Sinne von „Ohne Semantik“[55] bedeutungslos.

Um diesem Dilemma zu entkommen, muss eine Möglichkeit gefunden werden, die ausgezeichneten Textelemente bzw. die Annotationen soweit zu verschlüsseln, dass eine maschinenlesbare Verarbeitung erreicht und explizit gegebenes Wissen automatisch abgeleitet wird. Aus diesem Grund wurden auf Basis von XML die weiteren Sprachen *Resource Description Framework* (RDF), *Resource Description Framework Schema* (RDFS) und OWL vom W3C definiert.

Ohne RDF, RDFS und OWL müssten vorher die Bedeutungen der einzelnen benutzten Tags in einem Dokument untergebracht und festgehalten werden, damit ein problemloser Datenaustausch zwischen zwei Anwendungen stattfinden kann. Aus praktischer Sicht führt dies zu einer umfangreichen und weitschweifigen Dokumentation. Durch die Definition und Standardisierung von RDF, RDFS und OWL erübrigt sich die Erstellung solcher Dokumentationen.

Im Großen und Ganzen stellt XML einen grundlegenden Standard für das Semantic Web dar. Die Hauptaufgabe von XML besteht darin, strukturierte Informationen zu speichern und auszutauschen. Somit dient XML als syntaktische Grundlage für RDF, RDFS und OWL.

[55] [HKRS08], S. 30

3.2. RDF

Das *Resource Description Framework*, RDF, ist eine vom W3C standardisierte formale Sprache, die der Beschreibung strukturierter Informationen dient. Mit der Unterstützung von RDF können Anwendungen Daten im Web austauschen, ohne dass die Daten ihre ursprüngliche Bedeutung verlieren. Das Ziel von RDF ist nicht die korrekte Darstellung von Informationen, sondern die Weiterverarbeitung erhaltener Informationen. RDF ist eine Meta-Sprache, die u. a. verwendet wird, um Webinhalte beschreiben zu können.

Eine nicht minder wichtige Rolle spielen dabei die URI[56], mit denen man Ressourcen jeglicher Art, also nicht nur Webseiten, identifizieren kann. In seinem Semantic Web Artikel im „Scientific American“[57] beschreibt Tim Berners-Lee folgende Vision vom Semantic Web: Nach seiner Meinung wird in Zukunft das Semantic Web aus dem virtuellen Bereich ausbrechen und sich auf die wirkliche Welt ausdehnen. Mit Hilfe von URI kann man jegliche technische Ressourcen wie elektronische Geräte identifizieren, was bedeutet, dass beispielsweise Fernsehapparate oder Fernbedienungen mittels RDF beschrieben werden können. Solche mit RDF definierten Geräte teilen dann anderen Geräten ihre Funktionalitäten mit, so dass eine Interoperabilität zwischen den technischen Ressourcen besteht.

3.2.1. RDF: Die grundlegenden Ideen

An dieser Stelle muss klar gestellt werden, dass es sich bei RDF korrekterweise nicht um eine „Sprache“ sondern um ein Datenmodell[58] handelt. Während XML durch eine Baumstruktur dargestellt wird, basiert RDF auf einem graphenorientierten Datenschema. Durch XML kann man hierarchisch strukturierte Informationen darstellen, da i. d. R. die meisten elektronischen Informationen derartige Strukturen aufweisen. Im Gegensatz dazu wurde RDF nicht entwickelt, um einzelne Dokumente hierarchisch zu strukturieren, es beschreibt lediglich die Beziehungen zwischen den Ressourcen.

3.2.2. RDF in Bezug auf das Semantic Web

Im Semantic Web werden Informationen durch *Statements*[59] repräsentiert. Das *Statement* besteht aus einem Subjekt, einem Prädikat und einem Objekt[60]. Zusammenfassend bezeichnet man diese drei Komponenten auch als *Triple*. Diese drei Elemente eines

[56] Uniform Resource Identifier
[57] s. S. 21 K. 2.3
[58] [HKRS08], S. 36
[59] engl. für Aussagen
[60] [HFBL09], S. 68

Statement haben Bedeutungen ähnlich den Bedeutungen in der englischen Grammatik[61]. Ein Subjekt oder Objekt bezieht sich auf einen Sachgegenstand, der durch ein *Statement* beschrieben wird. Die Aufgabe des Prädikats ist es, Subjekte zu beschreiben und eine Verbindung mit dem Objekt herzustellen. Eine andere Bezeichnung für Prädikat ist der Begriff *Property*.

Mit Hilfe von RDF Dokumenten werden die Beziehungen zwischen dem *Triple* durch Graphen repräsentiert, also durch eine Menge von Knoten, die durch gerichtete Kanten miteinander verbunden werden. Die Knoten stellen die Subjekte bzw. Objekte dar, die Kanten sind die Prädikate. Konkreter betrachtet bezieht sich ein Subjekt oder Objekt auf die im vorherigen Kapitel erwähnten Ressourcen, z.B. wird das Subjekt „Webseite“ mit einem Objekt „Autor“ oder „Urheber“ mittels Prädikat in Verbindung gebracht.

An dieser Stelle sei noch auf die in Kapitel 3.2 erwähnten URI hingewiesen. Sie werden eingesetzt, um ein Subjekt bzw. Objekt oder in diesem Fall eine Ressource eindeutig zu identifizieren und um eine eindeutige Namensgebung zu gewährleisten.

Zum einen könnte es vorkommen, dass Ressourcen verschiedenartig benannt werden, obwohl beide Ressourcen sich ähneln. Zum anderen könnten versehentlich gleiche Bezeichner benutzt werden, die sich jedoch auf unterschiedliche Ressourcen beziehen. Durch die URI werden solche Namenskonflikte behoben.

Man betrachte folgendes Beispiel eines *Statement*, das anschließend mit Hilfe eines Graphen dargestellt wird: „Die Webseite http://www.tomgruber.org/ wird von Tom Gruber betrieben“.

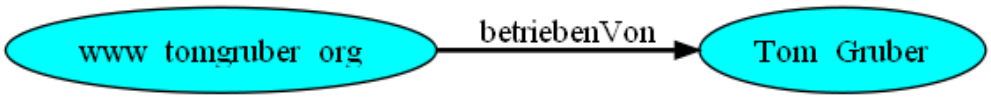


Abbildung 3.3: Graphendarstellung eines *Statement*

Eine Ressource repräsentiert in RDF ein Objekt oder auch ein Konzept, dessen Benennung durch eine URI erfolgt. Genauer gesagt referenziert eine URI auf tatsächlich gemeinte Dinge wie z. B. eine Webseite oder ein Buch[62]. Ausserdem werden die Prädikate ebenfalls mit URI versehen und somit sind Prädikate eine spezielle Form von Ressourcen[63]. Demzufolge muss die Abbildung 3.3 noch ein wenig korrigiert werden:

[61] S. P. O Regel
[62] [HFBL09], S. 71
[63] [AH08], S. 68

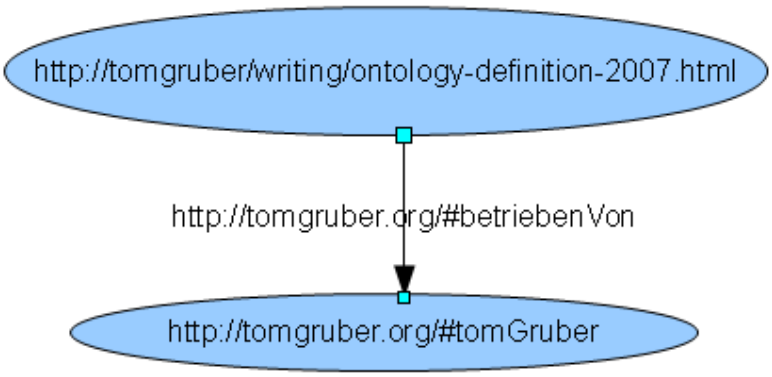


Abbildung 3.4: Graphendarstellung eines *Statement* inklusive URI

Wie bereits erwähnt werden die Ressourcen in RDF durch einen Knoten dargestellt. Darüber hinaus können Knoten auch Literale enthalten. Mit Literalen sind konkrete Datentypen wie *Integer* Zahlen oder Zeichenketten gemeint, sie können im Gegensatz zur Ressource niemals als Subjekt eines *Statement* dargestellt werden sondern ausschliesslich als Objekt[64]. Betrachtet man erneut das vorherige Beispiel, so kann man den Wert „Tom Gruber“ als Literal definieren und er ist somit das Objekt des *Statement*.

3.2.3. XML-Serialisierung von RDF

RDF Graphen eignen sich sehr gut um Informationen darzustellen, für die Speicherung sind sie jedoch zu abstrakt. Graphen sind für eine Analyse aus menschlicher Sicht zwar lesbar, andererseits sind sie ungeeignet für einen Datenaustausch zwischen verschiedenen Anwendungen[65]. Mit Hilfe der Serialisierung wird eine Möglichkeit geschaffen, abstrakte Datenmodelle wie die RDF Graphen in ein konkretes, speicherbares Format wie z. B. einen Bytestream oder eine Datei umzuwandeln, um den Austausch zwischen unterschiedlichen Anwendungen zu ermöglichen. Aktuell existieren verschiedene Methoden der Serialisierung von RDF Graphen. Die drei gängigsten Methoden sind: RDF/XML, RDF Triple Language[66] und N-Triple. RDF/XML ist die am weitesten verbreitete Methode. Dies liegt v. a. daran, dass jede übliche Programmiersprache die Verarbeitung von XML in Form von Programmbibliotheken unterstützt[67].

Die Syntax für RDF/XML erfolgt also durch eine XML basierte Schreibweise, so dass jeder übliche XML Parser RDF bzw. RDF/XML Dateien lesen und verarbeiten kann.

[64] [HFBL09], S. 519
[65] ebenda, S. 74
[66] Die genaue Fachbezeichnung lautet Turtle
[67] [HKRS08], S. 42

Die gegensätzlichen Datenmodelle von XML (Bäume) und RDF (Graphen) stellen keine Probleme dar. XML gibt einzig und allein die syntaktische Struktur des Dokuments vor, die einen RDF Graphen kodiert. Man darf dennoch nicht die Tatsache ignorieren, dass XML hierarchisch aufgebaut ist und aus diesem Grund die Kodierung des *Triple* hierarchisch erfolgen muss. Folgendes Beispiel beschreibt den vorherigen RDF Graphen[68]:

```
<?xml version="‘1.0"’ encoding="‘utf8"’?>
<rdf: RDF xmlns:rdf="‘http://www.w3c.org/1999/02/22-rdf-syntax-ns#"’
     xmlns:ex="‘http://tomgruber.org"’>
<rdf:Description rdf:about="‘http://tomgruber.org/writing/ontology-definition-2007.htm"’>
  <ex:betriebenVon>
    <rdf:Description rdf:about="‘http://tomgruber.org/#TomGruber"’>
    </rdf:Description>
  </ex:betriebenVon>
</rdf:Description>
<rdf:RDF>
```

Listing 5: RDF Code für den Graphen aus K. 3.2.2 Abb. 3.4

Nach der üblichen Angabe der XML Version wird das RDF Dokument mit einem Knoten vom Typ *rdf:RDF* eingeleitet, der allgemein als Wurzel eines RDF/XML Dokuments verwendet wird. Desweiteren werden die Namensräume *ex:* und *rdf:* deklariert, um Namenskonflikte zwischen Elementnamen zu verhindern. Darüber hinaus verweisen die Namensräume auf eine URI. Die *Triple* werden innerhalb des *rdf:RDF* Elements kodiert. Das Subjekt und das Objekt werden durch die Elemente vom Typ *rdf:Description* beschrieben und durch die Angabe des XML Attributs *rdf:about* erhält das Element eine Bezeichnung. Die Darstellung des Prädikats erfolgt durch das Element *ex:betriebenVon*, das im Namensraum <.. xmlns:ex=„http://tomgruber.org“> deklariert wird.

3.2.4. RDF Fazit

RDF wird hauptsächlich verwendet, um Metadaten auszudrücken und es beschreibt grundlegende Beziehungen zwischen zwei Einzelobjekten, die in diesem Fall durch Ressourcen repräsentiert werden. Das grundlegende Datenmodell von RDF ist graphenorientiert, mit ihm kann man Beziehungen visualisieren. Die Serialisierung erfolgt durch RDF/XML.

Nichts desto trotz birgt RDF einige Schwächen: RDF erlaubt nur Aussagen mit binären Prädikaten[69], es können also nur Beziehungen zwischen zwei Ressourcen abgebildet werden. Die Abbildung zur Aussage „X kennt Y“ wäre in RDF problemlos. Im Gegensatz

[68] s. S. 32 K. 3.2.2, Abb. 3.4
[69] [AH08], S. 69

dazu ist folgende Aussage mit RDF schwierig oder kaum beschreibbar: „X kennt Y und beide sind Dozenten der Informatik an der FH“. In der Realität jedoch verwendet man Prädikate mit mehr als zwei Subjekten bzw. Objekten.

Sicherlich könnte man dieses Problem mit Hilfe von RDFS[70], mit dem man eine einfache bis relativ komplexe Ontologie formalisieren kann, beheben. Mit RDFS ist es möglich, Klassen und Instanzen, mit deren Hilfe man Ressourcen „typisiert“[71], zu deklarieren. In RDFS repräsentieren Klassen stets eine Menge von Ressourcen, die durch URI identifiziert werden. Dennoch sind die Sprachmöglichkeiten von RDFS sehr eingeschränkt[72]. Einer der gravierendsten Nachteile von RDFS ist die Unmöglichkeit negative Aussagen auszudrücken[73] [74]. Desweiteren ist es in RDFS nicht möglich, den Subjekten bzw. Objekten gewisse Klassen unterschiedlich zuzuordnen. Z. B. schliessen sich die Klassen Mann und Frau gegenseitig aus[75]. Darüber hinaus fehlt bei RDFS die Möglichkeit, Kardinalitäten zwischen den Klassen anzugeben oder auch Klassen zu kombinieren.

Diese und andere diverse Schwächen und Defizite sind hauptverantwortlich für die Schwierigkeit, mit RDF bzw. RDFS ein semantisches Web zu erstellen. Die größten Nachteile liegen in der Darstellung komplexer Zusammenhänge, weshalb eine ausdrucksstarke Repräsentationssprachen benötigt wird.

[70] s. S. 30 K. 3.1.1
[71] Ressourcen bzw. Instanzen werden einem bestimmten Objekttyp zugeordnet
[72] [HKRS08], S. 118
[73] ebenda, S. 119
[74] In RDFS gilt prinzipiel die offene Welt Annahme, die besagt, dass nicht beweisbare Aussagen weder falsch noch wahr sind
[75] Disjunkte Klassen

3.3. OWL

OWL ist ein Akronym und steht für *Web Ontology Language*. Es ist seit dem Februar 2004 vom W3C als Ontologiesprache standardisiert. Aktuell ist OWL die am häufigsten verwendete Sprache, um Ontologien speziell im Semantic Web zu definieren.

Das Hauptaugenmerk von OWL liegt auf der Erstellung von Dokumenten, in denen OWL-Ontologien beschrieben werden. Um eine leichte Handhabung der Sprache zu erreichen und um einen unkomplizierten Umstieg von RDF auf OWL zu gewährleisten, baut OWL auf das RDF und RDFS Schema auf und verwendet die in Kapitel 3.2.3 vorgestellte RDF/XML als Syntax. Alle OWL Dokumente, die die RDF/XML Syntax beschreiben, sind auch gültige RDF Dokumente. Derartige OWL Dokumente nennt man auch OWL-RDF-Syntax[76].

Durch die Definition und Standardisierung von OWL wird den Problemen, die RDF und RDFS mit sich bringen, entgegengewirkt. Folgende Defizite werden mit OWL behoben:

- Definition von disjunkten Klassen[77]
- Boolsche Kombinationen von Klassen, u. a. mit Hilfe der Mengenlehre, also beispielsweise Vereinigung, Durchschnitt und Differenz
- Definition von Kardinalitäten, z. B. ein Kind hat genau zwei Elternteile
- Spezielle Ausprägungen von Eigenschaften, z. B. Transitivität, Einmaligkeit von Eigenschaften und Umkehrung einer Eigenschaft
- Umfang von Eigenschaften: In RDFS ist es nicht möglich, Wertebereiche zu definieren, die nur an bestimmten Klassen angewendet werden

OWL bietet somit weitaus mehr Möglichkeiten und Features an als RDF bzw. RDFS und ist darüber hinaus relativ ausdrucksstark.

3.3.1. OWL-Untersprachen

In OWL spielt die Balance zwischen Ausdrucksstärke der Sprache und effizientem Schlussfolgern eine bedeutende Rolle[78]. Diesen Balanceakt bezeichnet man auch als Skalierbarkeit, da folgender Aspekt nicht außer Acht zu lassen ist: Je ausdrucksstärker eine Sprache ist, desto höher ist ihre Komplexität, was zu schlechten Skalierbarkeitseigenschaften und

[76] [HKRS08], S. 42
[77] s. S. 35 K. 3.2.4
[78] [HKRS08], S. 126

ineffizientem Schlussfolgern führt. Aus diesem Grund definiert das W3C drei unterschiedliche Untersprachen, um die Auswahl zwischen den verschiedenen Ausdrucksstärken zu ermöglichen. Die drei OWL Untersprachen heißen: OWL Full, OWL DL[79] und OWL Lite. Folgende Abbildung veranschaulicht die Beziehungen zwischen den drei Untersprachen:

Abbildung 3.5: Beziehungen zwischen den drei OWL-Sprachen

Näher betrachtet ist OWL Lite eine Teilsprache von OWL DL, die wiederum eine echte Teilsprache von OWL Full ist.

OWL Full beinhaltet die Untersprachen OWL DL und OWL Lite und benutzt als einzige der drei Untersprachen uneingeschränkt alle RDFS Sprachelemente. Außerdem bietet OWL Full die maximale Kompatibilität zu RDF und ist darüber hinaus sehr ausdrucksstark. Doch wie bereits im vorherigen Kapitel erwähnt, steigt die Komplexität einer Sprache an, wenn sie zu ausdrucksstark ist, was u. a. zur Unentscheidbarkeit von OWL Full führt. Der Hauptgrund für diese Unentscheidbarkeit liegt v. a. in der Möglichkeit begründet, dass in OWL Full die Sprachkonstrukte frei vermischt werden können, d. h. es gibt keine strikte Trennung von Klassen, Eigenschaften, *Rollen*[80], Instanzen und Datentypen. So kann man beispielsweise den Namen einer Klasse auch als Instanz verwenden. Darüber hinaus werden bei OWL Full keine Unterschiede zwischen *ObjectProperty*[81] und *DatatypeProperty*[82] gemacht.

OWL DL ist im Gegensatz zu OWL Full bei der Verwendung von manchen RDFS Sprachelementen strikter eingeschränkt. V. a. werden die Typentrennungen strenger gehandhabt als bei OWL Full. Der Name einer Klasse darf also nicht gleichzeitig der Name

[79] OWL Description Logic
[80] s. S. 41 K. 3.3.2.2
[81] Objekteigenschaft
[82] Datentypeigenschaft

einer Instanz sein. Man muss klar zwischen Klassen, Individuen, abstrakten und konkreten *Rollen* unterscheiden. *ObjectProperty* und *DatatypeProperty* sind bei OWL DL nicht äquivalent. Die Sprachkonstrukte *inverseOf*, *TransitiveProperty*, *functionalProperty* sowie *symetricProperty*[83] können nicht für *DatatypeProperty* spezifiziert werden. Die Kardinalitätseinschränkungen dürfen nicht bei transitiven oder inversen Eigenschaften definiert werden. In Gegensatz zu OWL Full ist OWL DL entscheidbar, d. h. die Sprache wurde so entwickelt, dass aus einer Ontologie eine Aussage geschlussfolgert werden kann. OWL DL hält sich somit an die Richtlinien, die auf der *Description Logic* basieren.

OWL Lite unterstützt nur eine Teilmenge an Sprachkonstrukten und hat praktisch gesehen nur eine eingeschränkte Bedeutung. Hinter OWL Lite steckt ein einfaches, implementierendes Sprachfragment, welches die wichtigsten Sprachelemente beinhaltet. Bei OWL Lite sind folgende Sprachkonstrukte nicht erlaubt:

- *owl:unionOf*
- *owl:complementOf*
- *owl:disjointwith*

Im Gegensatz zu OWL Full und OWL DL ist OWL Lite wenig ausdrucksstark[84].

Weitere Einschränkungen können auf der W3C Webseite[85] nachgelesen werden. Die Untersprache OWL Lite eignet sich eher als Einsteigersprache, um sich in die Materie der Web Ontologie einzuarbeiten.

3.3.2. OWL: Syntax und Sprachkonstrukte

In diesem Kapitel wird auf die Ontologiesprache OWL eingegangen. Bei der späteren Implementierung des Projekts wird OWL für den Export der Daten in eine Ontologie eine Rolle spielen.

Genau wie bei RDFS besteht eine OWL Ontologie aus Klassen und *Properties*[86]. Allerdings kann man bei OWL die Beziehungen zwischen Klassen komplexer definieren, was bei RDFS nicht möglich ist[87].

3.3.2.1. OWL-Header

[83] Diese Sprachkonstrukte werden in K. 3.3.2.5 erläutert
[84] [HKRS08], S. 156
[85] `http://www.w3.org/TR/owl-features`
[86] s. S. 41 K. 3.3.2.2
[87] s. S. 35 K. 3.2.4

Eine OWL Ontologie wird in einem OWL Dokument abgelegt und ist ebenfalls ein gültiges RDF Dokument[88]. Die Beschreibung eines OWL Dokuments beginnt mit einem Ontologie Header und beinhaltet Informationen zu benutzten Namensräumen, Versionierung und Annotation. Diese Kopfdaten haben jedoch keine Auswirkung auf den Wissensgehalt der Ontologie. Da ein OWL Dokument auch als ein RDF Dokument betrachtet wird, hat jedes OWL Dokument eine Wurzel, in der die Namensräume spezifiziert werden, z. B.:

```
<rdf:RDF
  xmlns = "‘http://www.example.org"’
  xmlns:rdf= "‘http://www.w3.org/1999/02/22-rdf-syntax-ns#"’
  xmlns:xsd="‘http://www.w3.org/2001/XMLSchema#"’
  xmlns:rdfs="‘http://www.w3.org/2000/01/rdf-schema#"’
  xmlns:owl="‘http://www.w3.org/2002/07/owl#"’>
```

Listing 6: Beispiel eines OWL Header

Die erste Deklaration bezieht sich auf den allgemeinen Namensraum einer Ontologie. Durch den in der letzten Zeile angegebenen OWL Namensraum wird das OWL Vokabular eingeführt. Die anderen Namensräume, also *rdf*, *rdfs* und *xsd*, werden ebenfalls benötigt, da ein OWL Dokument auf den Konstrukten von RDF, RDFS und XML aufbaut.

Darüber hinaus kann man für die gesamte Ontologie weitere allgemeine Informationen angeben. Die Informationen werden dann innerhalb eines *owl:Elements* untergebracht. Folgendes Beispiel veranschaulicht dies:

```
<owl:Ontology rdf:about="‘"’> <rdfs:comment
rdf:datatype="‘http://www.w3.org/2001/XMLSchema#String"’>SWRC  Ontologie in der Version
    vom März 2009</rdfs:comment> <owl:versionInfo>v0.5</owl:versionInfo>
<owl:imports rdf:resource="‘http.//www.example.org/foo"’>
<owl:priorVersion rdf:resource="‘http://ontoware.org/projects/swrc"’ />
</owl:ontology>
```

Listing 7: OWL Header mit weiteren zusätzlichen Informationen

Zur Angabe der Version benutzt man folgende Elemente:

- *owl:versionInfo*
- *owl:priorVersion*

Aus Gründen der Vollständigkeit sei noch auf folgende Elemente hingewiesen, die ebenfalls zur Versionierung beitragen:

- *owl:backwardCompatibleWith*
- *owl:incompatibleWith*

[88] s. S. 37 K. 3.3

- *owl:DeprecatedClass*
- *owl:DeprecatedProperty*

Es kann vorkommen, dass Teile einer beschriebenen Ontologie noch unterstützt, jedoch nicht weiter verwendet werden sollen. Solche Teile einer Ontologie werden durch die Elemente *owl:DeprecatedClass* bzw. *owl:DeprecatedProperty* gekennzeichnet. Die anderen Elemente verweisen auf andere Ontologien.

3.3.2.2. Klassen, Rollen und Individuen

Eine OWL Ontologie besteht aus einer Reihe von Klassen bzw. OWL Klassen, aus denen man ggf. Individuen bildet. Genauer betrachtet handelt es sich bei Individuen um Instanzen von Klassen. Ein weiterer Grundbaustein von OWL sind die *Properties*[89] bzw. *OWL Properties*, die man aus RDF kennt. Die *Properties* stellen die Prädikate eines *Statement*[90] dar und beschreiben die Beziehungen zwischen Objekten. Eine andere Bezeichnung für *OWL Properties* ist der Begriff *Rollen*. Beide Begriffe werden als äquivalent angesehen und in dieser Arbeit gleichermaßen verwendet.
Eine einfache Klasse wird in OWL durch das Element *owl:Class* definiert.

```
1 <rdf:Description rdf:about="Professor">
2 <rdf:type rdf:resource="&owl;Class">
3 </rdf:Description>
```

Listing 8: Klassendeklaration in OWL

Äquivalent dazu kann man die Kurzform verwenden:

```
1 <owl:Class rdf:about="Professor" />
```

Listing 9: Kurzform der Klassendeklaration

Mit dem Ausdruck *rdf:about* wird der Klasse der Name „Professor“ zugeordnet. Alternativ dazu könnte man auch *rdf:id* verwenden.

In OWL gibt es zwei vordefinierte Klassen, nämlich *owl:Thing* und *owl:Nothing*. Die allgemeinste Klasse ist *owl:Thing*, sie beinhaltet alles, d. h. die gesamte ihr untergeordnete Ontologie. Im Gegensatz dazu ist die Klasse *owl:Nothing* leer. Demzufolge ist jede Klasse eine Unterklasse von *owl:Thing* und Superklasse von *owl:Nothing*.

Möchte man aus der vorherigen Klasse eine Instanz bilden, also ein Individuum deklarieren, geht man folgendermaßen vor:

[89] vgl. S. 32 K. 3.2.2
[90] s. S. 32 K. 3.2.2

```
<Professor rdf:about="Tom Gruber" />
```

Listing 10: Deklaration von Instanzen bzw. Individuen in OWL

Wie man an diesem Beispiel erkennt, werden Individuen als RDF Instanzen deklariert. Man unterscheidet in OWL zwischen zwei verschiedenen Arten von *Rollen*: Abstrakte und konkrete *Rollen*. Durch die Definition von abstrakten *Rollen* werden Individuen mit anderen Individuen verbunden. Im Unterschied dazu werden bei konkreten *Rollen* Individuen mit Datenwerten verknüpft. Mit Datenwerten sind die Elemente von XML Datentypen gemeint, also beispielsweise *xsd:integer* oder *xsd:string*. Die Deklaration von *Rollen* in OWL erfolgt ähnlich wie die Deklaration von Klassen. Bei abstrakten *Rollen* verwendet man *owl:ObjectProperty*[91] und bei konkreten *Rollen* das Element *owl:DatatypeProperty*[92]. *Rollen* beziehen sich i. d. R. auf bestimmte Bereiche und die Definition zu solchen Bereichen erfolgt durch *rdfs:domain* und *rdfs:range*, wie das folgende Beispiel veranschaulicht:

```
<owl:ObjectProperty rdf:about="gelehrtVon">
  <rdfs:domain rdf:resource="Kurs" />
  <rdfs:range rdf:resource="Fakultätsmitglied"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="Kursname">
  <rdfs:domain rdf:resource="Kurs"/>
  <rdfs:range rdf:resource="&xsd;string">
</owl:DatatypeProperty>
```

Listing 11: Deklaration von abstrakten und konkreten *Rollen* in OWL

Im ersten Beispiel wird festgelegt, dass ein Kurs nur von einem Fakultätsmitglied gelehrt wird. Dies erfolgt durch *owl:ObjectProperty*, das die beiden Individuen „Kurs“ und „Fakultätsmitglied“ miteinander in Verbindung setzt. Möchte man dem Kurs einen Namen geben, geschieht dies durch *owl:DatatypeProperty*, das einem Individuum einen konkreten Wert zuordnet, also in diesem Fall einen Wert vom Typ *String* bzw. *xsd:string*.
Durch *rdfs:Domain* bezieht sich eine Rolle auf ein bestimmtes Subjekt und mit *rdfs:range* wird festgelegt, dass auf ein Prädikat bzw. *Property* nur ein bestimmter Wert folgen darf.

3.3.2.3. Klassenbeziehungen

Die einfachste Klassenbeziehung in OWL wird mittels *rdfs:subClassOf* definiert, wie das folgende Beispiel veranschaulicht:

[91] vgl. S. 37 K. 3.3.1
[92] vgl. ebenda

```
<owl:Class rdf:about="Professor">
<rdfs:subClassOf rdf:resource="Fakultätsmitglied" />
</owl:Class>
```

Listing 12: Einfache Klassenbeziehung durch *subClassOf* in OWL

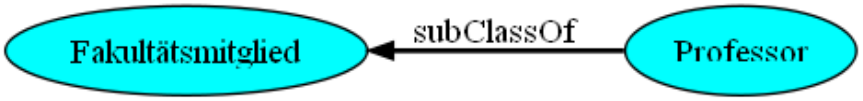


Abbildung 3.6: Graphendarstellung von *subClassOf*

Dieses Beispiel besagt, dass „Professor“ eine Unterklasse von „Fakultätsmitglied“ ist. Das Sprachkonstrukt *rdfs:subClassOf* gilt in OWL als transitiv. Hierzu betrachte man als Beispiel folgende transitive Beziehung zwischen den Subjekten A, B und C:

$$A \rightarrow B \tag{1}$$

$$B \rightarrow C \tag{2}$$

$$\Rightarrow A \rightarrow C \tag{3}$$

Klasse A ist eine Unterklasse von B und B wiederum ist eine Unterklasse von C. Demzufolge ist auch A eine Unterklasse von C. Durch die Transitivität wird die Möglichkeit geschaffen, Schlussfolgerungen aus implizitem Wissen herzuleiten.
Folgendes OWL Beispiel stellt die Schlussfolgerung dar, dass ein Professor auch eine Person ist:

```
<owl:Class rdf:about="Professor">
<rdfs:subClassOf rdf:resource="Fakultätsmitglied" />
</owl:Class>
<owl:Class rdf:about="Fakultätsmitglied">
<rdfs:subClassOf rdf:resource="Person" />
</owl:Class>
```

Listing 13: OWL Beispiel zur Transitivität

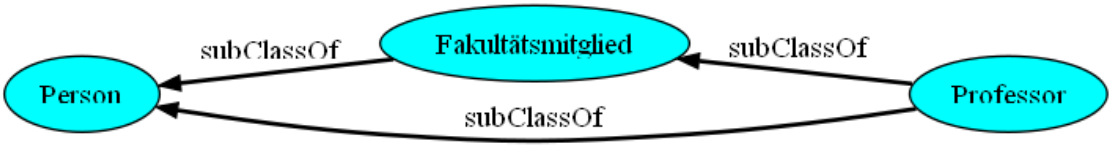


Abbildung 3.7: Graphendarstellung des OWL-Quellcode aus Listing 13

Zwei Klassen können aber auch disjunkt zueinander stehen, d. h. die Individuen beider Klassen haben keine Gemeinsamkeiten. In OWL werden solche Beziehungen mit Hilfe von *owl:disjointwith* deklariert. Hierzu ein erneuter Verweis auf das Mann-Frau-Beispiel[93]:

```
<owl:class rdf:about="Mann">
<owl:disjointwith rdf:resource="Frau" />
</owl:class>
```

Listing 14: OWL Beispiel zur Disjunktheit von zwei Klassen

Eine andere Beziehungsart ist die Gleichartigkeit von zwei Klassen, die mittels *owl:equivalentClass* deklariert wird. Wenn Klassen Unterklassen voneinander sind, so sind sie gleich.

```
<owl:Class rdf:about="Publikation">
  <owl:equivalentclass="Publication" />
</owl:Class rdf:about="Publikation">
```

Listing 15: OWL Beispiel zu äquivalenten Klassen

Abbildung 3.8: Graphendarstellung von äquivalenten Klassen

Die Klasse „Buch“ ist sowohl Unterklasse von „Publikation“ als auch von „Publication“. Die Sprachkonstrukte *owl:disjointwith* und *owl:equivalentClass* tragen ebenfalls dazu bei, Inferenzen[94] aus implizitem Wissen herzuleiten.

3.3.2.4. Komplexe Klassenbeziehungen und Definitionen durch logische Konstruktoren

Die vorherigen Beispiele zeigten einige Sprachelemente auf, mit denen man in OWL einfache Ontologien aufbauen kann. Bei genauerem Hinsehen jedoch erkennt man, dass die Ausdrucksstärke noch zu wünschen übrig lässt und nur wenig über RDFS hinausgeht.
Eine höhere Ausdrucksstärke wird in OWL erst durch die Verwendung von logischen Konstruktoren, wie Konjunktion, Disjunktion und Negation erreicht. Klassen können also durch ein logisches *und*, *oder* und *nicht* miteinander in Beziehung gesetzt werden und somit ist es möglich, komplexere Beziehungen zu definieren. Die OWL Sprachelemente zu

[93] s. S. 35 K. 3.2.4
[94] Inferenz ist eine andere Bezeichnung für Schlussfolgerung

den logischen Konstruktoren heißen: *owl:intersectionOf*, *owl:unionOf* und *owl:complementOf*.
Mit Hilfe solcher Sprachkonstrukte können in OWL folgende Aussagen abgebildet werden:

Schnittmenge bzw. logisches *und*

$$KlasseA = \{1, 2, 3, 4, 5\} \tag{4}$$
$$KlasseB = \{1, 8, 7, 5, 3\} \tag{5}$$
$$KlasseC = A \cap B = \{1, 3, 5\} \tag{6}$$

In OWL übertragen wird diese Schnittmenge folgendermaßen aussehen:

```
<owl:Class rdf:ID="C">
 <owl:intersectionOf rdf:parseType="Collection">
 <owl:Class rdf:about="#A" />
 <owl:Class rdf:about="#B" />
 </owl:intersectionOf>
</owl:Class>
```

Listing 16: Schnittmenge in OWL

Konkret heißt dies, dass C genau alle Objekte enthält, die die Klasse A und die Klasse B gemeinsam haben.

Vereinigung bzw. logisches *oder*

$$KlasseA = \{1, 2, 3, 4, 5\} \tag{7}$$
$$KlasseB = \{1, 8, 7, 5, 3\} \tag{8}$$
$$KlasseC = A \cup B = \{1, 2, 3, 4, 5, 7, 8\} \tag{9}$$

In OWL lautet diese Vereinigung:

```
<owl:Class rdf:ID="C">
 <unionOf rdf:parseType="Collection">
 <owl:Class rdf:about="#A" />
 <owl:Class rdf:about="#B" />
 </owl:unionOf>
</owl:Class>
```

Listing 17: Vereinigung in OWL

Diese Vereinigung besagt, dass C alle Objekte enthält, die in Klasse A und in Klasse B enthalten sind.

Negation bzw. logisches *nicht*

$$KlasseA = \{1, 2, 3, 4, 5\} \tag{10}$$

$$KlasseB = \{1, 8, 7, 5, 3\} \tag{11}$$

$$KlasseC = A \setminus B = \{2, 4\} \tag{12}$$

In OWL lautet die Negation:

```
<owl:Class rdf:about="A">
 <rdfs:subClassOf>
 <owl:complementOf rdf:resource="B"/>
</rdfs:subClassOf>
</owl:Class>
```

Listing 18: Negation in OWL

Die Differenz bezieht sich zu meist auf zwei Mengen bzw. auf zwei Klassen. Die Negation besagt, dass Klasse C, die Objekte enthält, die in Klasse A aber nicht in Klasse B enthalten sind. Das Sprachkonstrukt *owl:complementOf* hat dieselbe Bedeutung wie *owl:disjointwith*. Alternativ kann man das obige Beispiel auch mit einem *owl:disjointwith*[95] ausdrücken.

An dieser Stelle sei noch darauf hingewiesen, dass die logischen Konstruktoren nicht bei OWL Lite verwendet werden können[96].

Mit den logischen Konstruktoren von OWL ist man dazu in der Lage, auch Mengenoperationen auf Basis der Klassen bzw. Klassenbeziehungen anzuwenden, was bei RDFS nicht möglich ist.

3.3.2.5. Rollen-Einschränkungen und - Eigenschaften

Der Vollständigkeit halber wird an dieser Stellle noch kurz auf die Rolleneigenschaften und Rolleneinschränkungen eingegangen.

In OWL ist es auch möglich, zu den *Properties* bzw. *Rollen*, Einschränkungen festzulegen. So z. B. wenn man ausdrücken möchte, dass eine Klausur maximal zwei Prüfer haben kann oder eine Klausur mindestens drei Themengebiete umfasst. Die Definitionen zu solchen Einschränkungen erfolgen in OWL durch *owl:maxCardinality* bzw.

[95] vgl. S. 42 K. 3.3.2.3, List. 14
[96] s. S. 37 K. 3.3.1

owl:minCardinality. Es besteht also die Möglichkeit Kardinalitäten zu definieren.
Zu *Rollen* können darüber hinaus noch weitere Eigenschaften bestimmt werden. In OWL existieren vier Arten von Rolleneigenschaften, deren Sprachkonstrukte folgendermaßen aussehen[97]:

- *owl:transitiveProperty*: definiert eine transitive *Property*[98]
- *owl:symetricProperty*: definiert eine symmetrische *Property*, die folgendes besagt: Stehen A und B in symmetrischer Rollenbeziehung, dann steht auch B zu A in Beziehung
- *owl:functionalProperty*: definiert eine funktionale *Property*, d. h. zu jedem Objekt existiert mindestens ein konkreter Wert
- *owl:InverseFunctionalProperty*: definiert eine inverse *Property*, die besagt, dass zwei völlig unterschiedliche Objekte nicht denselben Wert haben können. Genauer: Die *Property* „istProjektleiterFür“ ist die inverse Funktion zu „hatProjektLeiter“.

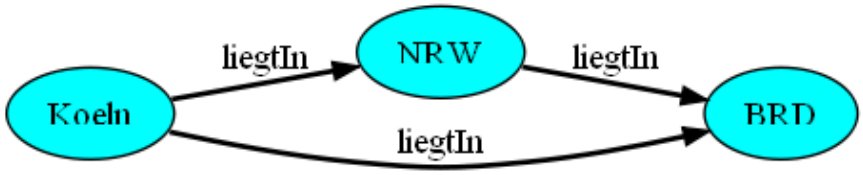


Abbildung 3.9: *Property* „LiegtIn“ ist transitiv: Köln liegt in NRW und NRW liegt in der BRD. Daraus folgt, dass Köln in der BRD liegt

Abbildung 3.10: *Property* „hatKollegen“ ist symmetrisch. Wenn ProfessorA den ProfessorB zum Kollegen hat, dann gilt dies auch umgekehrt

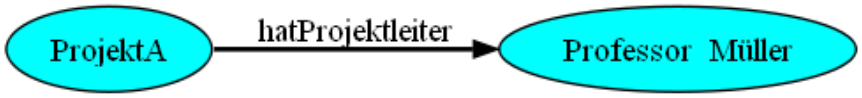


Abbildung 3.11: *Property* „hatProjektleiter“ ist funktional, d. h. auf das Objekt ProjektA folgt genau der konkrete Wert „Professor Müller“

[97] [AH08], S. 126
[98] vgl. S. 42 K. 3.3.2.3

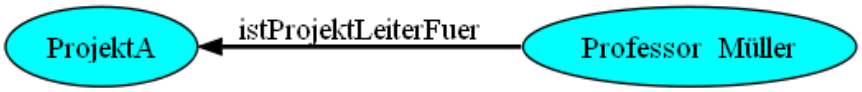


Abbildung 3.12: *Property* „istProjektleiterFuer“ ist die inverse Funktion zu „hatProjektleiter“

Die Verwendung von Rolleneigenschaften ist in OWL DL und OWL Lite nur eingeschränkt nutzbar[99].

3.3.3. OWL Fazit

In den vorherigen Kapiteln wurden einige Grundlagen zur OWL Syntax und zu Sprachelementen vorgestellt. Mit OWL ist man in der Lage, im Gegensatz zu RDF bzw. RDFS, ein relativ komplexes Wissen zu modellieren. Desweiteren ist OWL eine von dem W3C standardisierte Web-Ontologie Sprache und darüber hinaus auch eine Recommendation des W3C. Das hat den Vorteil, dass es aktuell etliche Werkzeuge und Programme gibt, die die Entwicklung und Erstellung von OWL unterstützen.

Doch OWL hat darüber hinaus mehr Möglichkeiten, Wissen noch intelligenter zu modellieren – sogar weitaus mehr Möglichkeiten als in den vorherigen Kapiteln beschrieben. Auf eine weitere Vertiefung dieser Thematik wird an dieser Stelle jedoch verzichtet, da diese den Rahmen dieser Diplomarbeit sprengen wurde.

Die praktische Anwendung von OWL wird im nächsten Kapitel anhand der Marktstudienkategorien vorgestellt.

[99] s. S. 37 K. 3.3.1

4. Entwicklung einer Kategorienontologie mit OWL

Für die spätere Implementierung ist es von grundlegender Bedeutung, zunächst eine Kategorienontologie zu entwickeln, da es u. a. eine Zielsetzung ist, die Marktstudien einer oder mehreren Kategorien zuzuordnen. Ohne eine dazugehörige Kategorieontologie wäre es schwierig, auf Basis von OWL eine Kategorienzuordnung vorzunehmen. Als Basis für die Erstellung einer Kategorienontologie werden die Kategorien aus dem Marktstudienportal verwendet. Dieser Schritt ist nicht nur notwendig, sondern auch ausgesprochen praktisch, da die Kategorien und ihre Bezeichnungen ja schon vorhanden sind und es deswegen nicht mehr nötig ist, eine von Grund auf neue Ontologie zu entwickeln.

Vor der Entwicklung der Grundontologie wird zunächst die der Entwicklung zugrundeliegende Motivation erläutert, anschließend wird diese mit der Zielsetzung dieses Projekts in Verbindung gesetzt. Als nächster Schritt muss entschieden werden, welche OWL Untersprachen bei der Erstellung der Grundontologie benutzt werden sollen[100].

Für die Marktstudie selbst muss ebenfalls eine Ontologie konzipiert werden, die dann im Anschluss an die entwickelte Kategorienontologie beschrieben und vorgestellt wird.

4.1. Motivation und Zielsetzung

Nun stellt sich die Frage, wozu man eine Ontologie für Marktstudien braucht und welchem Zweck sie später dienen werden. Neben der Erweiterung des Marktstudienportals in das Semantic Web wird ein anderes basales Ziel verfolgt: Mit Hilfe der Ontologie Sprache OWL möchte man Marktstudien in verwandtschaftliche Beziehungen zueinander setzen. Über die Kategorisierung wird eine einfache und praktikable Methode geschaffen, eine verwandtschaftliche Beziehung zwischen den Marktstudien zu erreichen.

Die in der Projektbeschreibung generierte OWL Datei beinhaltet die Marktstudienontologie mitsamt ihrer Kategorienzuordnung. Die OWL Datei stellt somit das Endprodukt nach einer erfolgreichen *Text Extraction* dar. Diese Datei wird dann einer semantischen Suchmaschine zur Verfügung gestellt. Solche Suchmaschinen besitzen die Fähigkeit, OWL Dateien zu lesen und zu interpretieren sowie ggf. daraus Schlussfolgerungen zu ziehen, sofern sie auf Basis des W3C Standards arbeiten. Ein Beispiel für eine semantische Suchmaschine ist das Projekt Swoogle[101], das von der University of Maryland Baltimore County entwickelt wurde. Aber auch andere Hersteller von Suchmaschinensoftware

[100] s. S. 37 K. 3.3.1
[101] `http://swoogle.umbc.edu/`

wie z. B. Apache mit seinen Produkten Lucene[102] und Solr[103], haben semantische Suchmaschinen auf Basis des W3C Standards entwickelt. Die semantischen Suchmaschinen müssen dann so eingestellt werden, dass sie verwandtschaftliche Beziehungen erkennen und angezeigen.

Mit der Zeit wird das OWL Dokument immer mehr mit Marktstudienontologien gefüllt, so dass diese Datei später als eine Art Wissensrepräsentation für Marktstudien dient.

4.2. Auswahl einer OWL-Untersprache

Wie in Kapitel 3.3.1 bereits beschrieben stehen drei OWL Untersprachen zur Auswahl: OWL Full, OWL DL und OWL Lite. Welche Untersprache bevorzugt wird, hängt von der späteren Verwendung der Ontologie ab. In Bezug auf die Zielsetzung dieses Projekts spielt die Inferenz von Wissen eine große Rolle. Die zu verwendende Ontologie sollte so entwickelt und aufbereitet werden, dass explizites Wissen abgeleitet werden kann. Bei der Suche nach dem Begriff „Apfel“ z. B. soll unmittelbar erkannt werden, dass es sich hier um eine Frucht handelt und eine verwandtschaftliche Beziehung zum Begriff „Birne“ besteht. Aus diesem Grund kommt OWL Lite als Untersprache nicht in Frage – es ist wenig ausdrucksstark und darüber hinaus untestützt es nicht alle OWL Sprachelemente[104]. Doch auch OWL Full wäre keine gute Alternative: Der Aspekt der hohen Komplexität und die daraus folgende Unentscheidbarkeit[105] dieser Untersprache spricht gegen eine Verwendung von OWL Full. Übrig bliebe demzufolge noch OWL DL, das im Gegensatz zu OWL Full entscheidbar ist und dadurch die Ableitung von Inferenzen zulässt. In Anbetracht dieser Vorteile wird innerhalb des hier vorliegenden Projekts die Entwicklung der Grundontologie auf OWL DL basieren. Weitere Vorzüge von OWL DL werden später während der Ontologie Entwicklung herausgestellt. Im Hinterkopf zu behalten ist jedoch, dass in OWL DL die Nutzung von Rolleneigenschaften nur beschränkt möglich ist[106].

4.3. Werkzeuge zur Modellierung von Ontologien

Es ist nicht empfehlenswert, eine Ontologie mittels eines herkömmlichen Editors zu erstellen, da man auf diese Art und Weise den Überblick verloren würde. Wie bereits erwähnt handelt es sich bei OWL um eine vom W3C standardisierte Ontologienprache. Aus

[102] http://lucene.apache.org/
[103] http://lucene.apache.org/solr/
[104] s. S. 37 K. 3.3.1
[105] ebenda
[106] s. S. 46 K. 3.3.2.5

diesem Grund, existieren aktuell viele Werkzeuge und Softwareapplikationen, die die Erstellung von Ontologien unterstützen. Eine dieser Anwendungen heißt *Protege*, sie wurde von der Stanford University entwickelt. In der Literatur zum Thema Ontologie und OWL wird dieses Programm häufig für die Modellierung von Ontologien empfohlen[107] [108] [109].

Bei *Protege* handelt es sich um eine *Java* basierte Open Source Software die demzufolge für jede Plattform frei zur Verfügung steht. Die Software verfügt über eine grafische Benutzeroberfläche, was einen guten Überblick über alle erstellten Klassen, *Rollen* und Individuen ermöglicht und eine manuelle Eingabe der Klassenbeziehungen untereinander überflüssig macht. Für die Erstellung einer neuen Ontologie benötigt *Protege* lediglich eine URI, die auf die Ontologie referenziert.

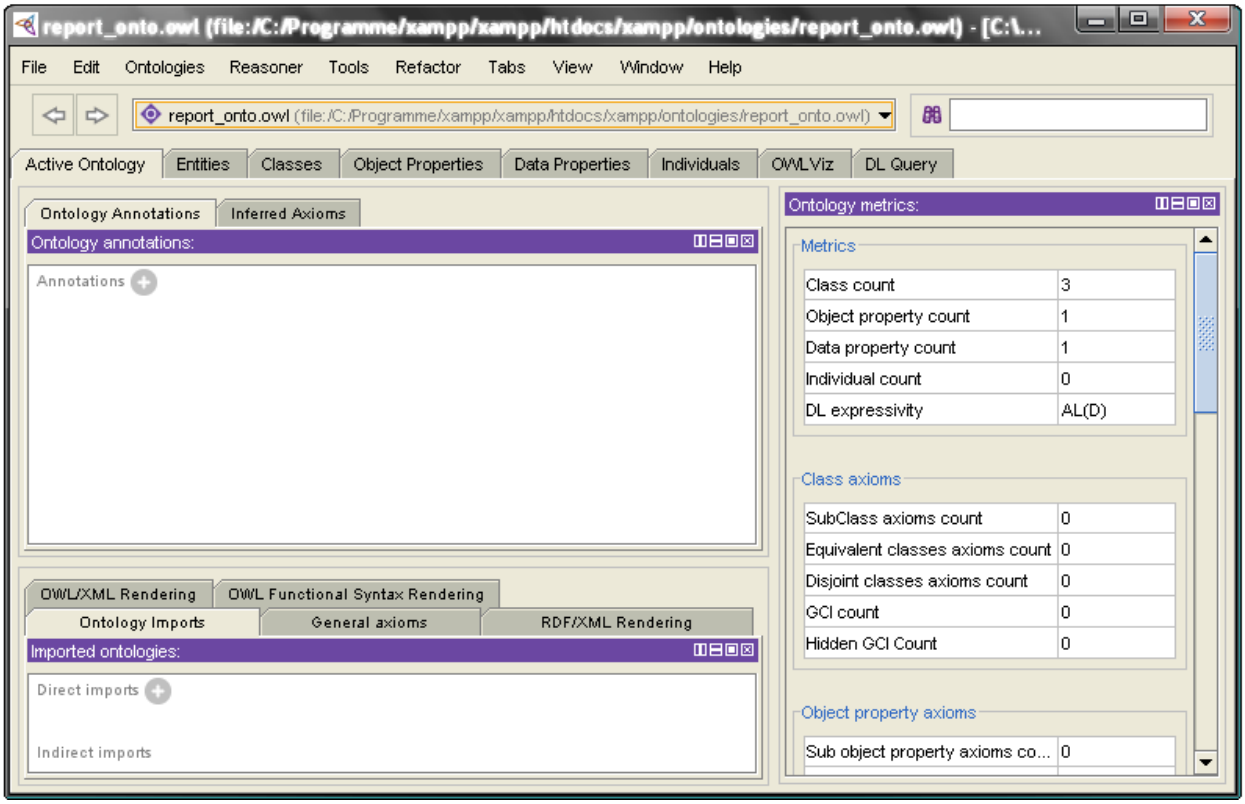


Abbildung 4.1: Programmoberfläche von *Protege*

Protege kann ähnlich wie die Entwicklungsumgebung *Eclipse* um weitere Plugins erweitert werden. So kann man das Programm *Graphviz* in Protege integrieren. *Graphviz* ist eine Anwendung zur Visualisierung von Graphen, mit Hilfe dieses Plugins kann eine Ontologie durch einen Graphen dargestellt werden.

[107] s. [AH08]
[108] s. [HFBL09]
[109] s. [SEBT09]

4.4. Die Marktstudienkategorien

Die Kategorien sind ein fester Bestandteil des Shopsystems *osCommerce* und werden aktuell in der zugehörigen MySQL Datenbank gespeichert. Die Beziehungen zwischen den Kategorien werden nach dem „Vater-Sohn-Prinzip“ hierarchisch aufgebaut. Es gibt immer eine Root Kategorie, die eine oder mehrere Kindkategorien besitzt, die Kindkategorie selbst besitzt maximal eine Unterkategorie. Von der Wurzel ausgehend beträgt die maximale Anzahl der Hierarchieebenen drei. Der hierarchische Aufbau der Kategorien basiert auf einer Baumstruktur. Eine andere fachspezifische Bezeichnung dafür ist der Begriff „Kategorienbaum“ [110]. Folgendes Beispiel zeigt den Kategorienbaum zur Kategorie „Automotive/Transport“[111] aus der englischsprachigen Version des Marktstudienportals:

- Automotive / Transport (1. Ebene)
  - Automotive (2. Ebene)
    - Trucks (3. Ebene)
    - Motorcycles
    - Cars
    - Insurance / Financing
    - Accessories / Spare Parts / After Sales Market
    - Bycicles
    - General Automotive
  - Rail and Transport
  - Logistics / Shipping
  - Air Transport
  - General Automotive and Transport

Die Unterkategorie „Automotive“ besitzt selber weitere Kindkategorien. Das dazugehörige Tabellenschema sieht folgendermaßen aus:

[110] Unternehmensintern verwendet man vornehmlich diesen Begriff
[111] `http://www.reports-research.com/market-surveys/cat_sitemap.php`

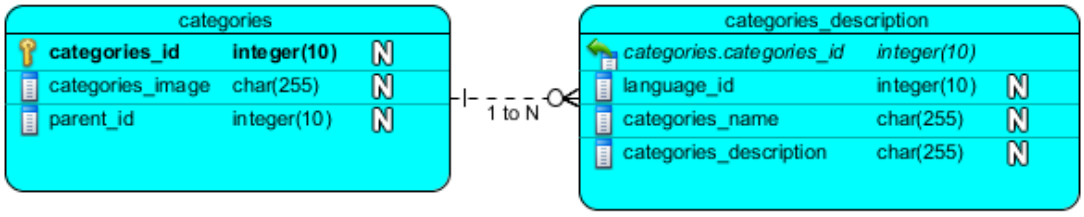


Abbildung 4.2: ERD-Diagramme von der Tabelle *categories* und *categories_description*

Für die Kategorien werden zwei Tabellen verwendet. Der Grund für diese Aufteilung liegt darin, dass das Marktstudienportal neben der deutschen Version, auch ein englisch-[112] und ein spanischsprachiges Pendant[113] anbietet, d. h. für jede der drei Sprachen müssen die Kategoriennamen angepasst werden. In der Tabelle *categories_description* bezieht sich die Spalte „language_id“ auf die *language* Tabelle der MySQL Datenbank. In der Tabelle *Categories* ist die Spalte „parent_id“ von Interesse. Über diese „parent_id“ wird eine Kategorie einer Wurzel bzw. einer Oberkategorie zugeordnet. Falls die Kategorie selbst die Wurzel ist, so wird in der „parent_id“ eine „0“ eingetragen.

4.5. Umsetzungskriterien für eine Ontologie

Auf Basis der Kategorien Tabellen aus der Datenbank können jetzt die Ontologien zu den Kategorien erstellt werden. Hier gibt es keinen vorgeschriebenen Weg, der angibt wie man eine Ontologie erstellen soll[114]. Die gängigste Methode ist das iterative Verfahren, d. h. man beginnt mit groben arbeitschritten und anschließend wird die zu entwickelnde Ontologie schrittweise verfeinert. Dennoch gibt es für die Modellierung von Ontologien folgende drei Empfehlungen[115]:

1. Alternative und brauchbare Wege prüfen – also nicht immer auf einem Lösungsweg beharren.
2. Die Entwicklung einer Ontologie unterliegt einem iterativen Prozess
3. Die in der Ontologie definierten Konzepte müssen sowohl logisch als auch physisch auf realen Objekten basieren. In diesem Fall beziehen sich die Konzepte auf die vorhandenen Kategorien

[112] `www.reports-research.com`
[113] `www.estudio-mercado.es`
[114] s. [NOY08]
[115] ebenda

4.6. Umstieg auf einen neuen Kategorienbaum

Bei den ersten Schritten zur Umsetzung der Ontologie für die Kategorien traten Probleme auf – v. a. bezüglich des aktuellen Bestandes der Kategorien und ihrer Beziehungen zu den Marktstudien.

Die aktuellen Kategorien im Marktstudienportal stammen noch aus der Gründungszeit des Unternehmens und wurden seitdem nicht mehr verändert, also auch nicht erweitert. Damals orientierte man sich an dem Konkurrenten marketresearch.com und übernahm im Großen und Ganzen seine Kategorisierung. Mittlerweile hat marketresearch.com jedoch seine Kategorien mitsamt der Struktur verändert und es ist nicht ersichtlich, auf welcher Grundlage die neuen Kategorien basieren. Aus diesem Grund entschied sich die dytec GmbH einen neuen Kategorienbaum zu erstellen. Die aktuellen Kategorien erwiesen sich als veraltet und aus wirtschaftlicher sowie branchenspezifischer Sicht nicht repräsentativ genug, da sie seit damals kaum verändert wurden.

Marktstudien sind i. d. R. branchenspezifisch und die Branchen werden im Portal durch Kategorien repräsentiert. An dieser Stelle sei nochmal darauf hingewiesen, dass Benutzer meistens Suchmaschinen verwenden, um ein bestimmtes Produkt bzw. in diesem Fall eine Marktstudie zu finden. Es liegt selbstverständlich im Interesse eines E-Commerce Unternehmens, dass seine Produktsortimente bei einer Suche immer ganz oben auf der Ergebnisliste der Suchmaschinen aufgeführt werden. Um dieses Ziel zu erreichen, müssen sogenannte Keywordanalysen vorgenommen werden. Dies ist z. B. im Bereich des SEO eine übliche Vorgehensweise, um den Trend zu erkennen, welche Begriffe zu diversen Branchen am häufigsten gebraucht und gesucht werden. Für solche Analysen bietet Google die Suchmaschinen Tools „Google Keyword Tool“[116], „Google Trends“[117] und „Google Insights for Search“[118] an. Mit diesen Analysen ist es möglich, aktuelle Markt- und Branchentrends zu beobachten. Bei der Entwicklung neuer Kategorien ist es aus marktwirtschaftlicher Sicht von Vorteil, auch auf die aktuellen Suchtrends und Markt- bzw. Branchentendenzen zu achten.

In den Jahren nach der Gründung des Marktstudienportals fehlten diese Keyword- und Trendanalysen, was dazu führte, dass die aktuellen Kategorien im Portal aus wirtschaftlicher und branchenspezifischer Sicht nicht repräsentativ genug sind.

Generell findet bei der Erweiterung von E-Commerce Anwendungen in das Semantic Web „eine starke Auswirkung auf unternehmerischer Seite statt, vor allem in Bezug auf

[116] https://adwords.google.de/select/KeywordToolExternal
[117] http://www.google.de/trends
[118] http://www.google.com/insights/search/?hl=de

Organisation, Management, Marketing, Vertrieb und natürlich die Positionierung und Sortimentgestaltung“[119]. In diesem Fall müssen das Marktstudiensortiment und die Kategorien umgestaltet und verbessert angeboten werden, um erfolgreich im Semantic Web Fuß zu fassen und eine zugehörige repräsentative Ontologie zu schaffen. Das Ziel muss also sein, die Marktstudien durch wohl durchdachte Kategorien zu klassifizieren, so dass zum einen die Auffindbarkeit der Marktstudien verbessert und zum anderen die Wettbewerbsfähigkeit des Marktstudienportals verstärkt wird.

Ein anderer Grund, der gegen die Verwendung der aktuellen Kategorien als Basisontologie spricht, ist die Masse an Marktstudien, die jede Kategorie beinhaltet. Es existieren zu viele Informationen, so dass zahlreiche Studien in der Masse untergehen und nicht mehr direkt gefunden werden können. Beim Semantic Web geht es darum, diese Informationsflut besser zu organisieren, so dass aus Sicht der Anwender nützliche Informationen herausgefiltert werden können. Momentan kommt es zumeist vor, dass eine Unterkategorie mehr Marktstudien beinhaltet als ihre Vaterkategorie. Für eine bessere Organisation der Informationen wäre es von Vorteil, die Unterkategorien noch einmal zu fragmentieren, also weitere Unterkategorien zu definieren, so dass die Auflistung aller Marktstudien und ihrer Kategorien übersichtlicher wird. Die gegenwärtig starre 3-Ebenen-Struktur der Kategorien[120] muss durchbrochen und erweitert werden, damit der Kategorienbaum eine höhere Flexibilität aufweist.

Als Ausgangspunkt für die Entwicklung einer neuen Kategorie verwendet man den von dem Unternehmen Portalix[121] zur Verfügung gestellten Kategorienbaum. Bei Portalix handelt es sich um einen engen Geschäftspartner der dytec GmbH, der auf die Klassifizierung von Internet-Domainnamen spezialisiert ist. Alle Begriffe aus dem neuen Kategorienbaum stammen nämlich aus dem Bereich der Klassifizierung von Domainnamen bzw. Keywords. Allerdings reicht der Kategorienbaum von Portalix als Basis nicht aus, da die Portalix-Begriffe aus der Klassifizierung von Domainnamen stammen und wenig markt- und branchenspezifisch sind. Als weitere Orientierungshilfen werden deshalb der neue Kategorienbaum von marketresearch.com sowie zusätzlich der aktuelle Kategorienbaum aus dem Marktstudienportals hinzugezogen. Auf Basis dieser drei Kategorienbäume – Portalix, marketresearch.com und markt-studie.de – wird der neue Kategorienbaum entwickelt. Anhand der Ähnlichkeiten von Begriffen werden einige Kategorien zusammengeführt und überflüssige entfernt. Der Vorgang findet manuell statt.

[119] s. [WAC08]
[120] s. S. 52 K. 4.4
[121] http://portalix.com/de/

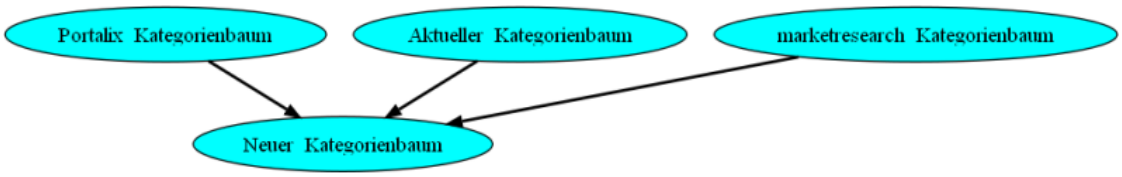


Abbildung 4.3: Zusammensetzung des neuen Kategorienbaums

Als Ergebnis erhält man einen sehr umfassenden Kategorienbaum, der im Großen und Ganzen alle gängigen Kategorien bzw. Branchen abdeckt. Es stellt sich dabei die Frage, welche Kategorien letztendlich übernommen werden sollen bzw. welche Kategorien zum aktuellen Bestand der Marktstudien thematisch passen.

4.6.1. Bereinigung des neuen Kategorienbaums

Anhand der in dem Portal vorhandenen Marktstudien muss eine Bestandsanalyse vorgenommen werden, in der geprüft wird, welche Marktstudien zu welcher neuen Kategorie passen. Damit man nicht vollständig die Übersicht verliert, werden für dieses Projekt nur die englischsprachigen Marktstudien[122] [123] aus dem Portal entnommen. Ziel ist es, unter Zuhilfenahme der entnommenen englischen Marktstudien, eine abgespeckte Version des neuen Kategorienbaums zu erstellen. Der Kategorienbaum muss noch einmal manuell bearbeitet werden. Vor der manuellen Erstellung des neuen Kategorienbaums werden folgende technische Schritte durchgeführt:

1. Speichern des neuen Kategorienbaums in einer eigens zum Testen entwickelten MySQL Datenbank
2. Auswahl aller englischen Marktstudien und Speicherung in der gleichen Test Datenbank

Zu Punkt 1 sei noch hinzugefügt, dass die Struktur des neuen Kategorienbaums auch auf dem „Vater-Sohn-Prinzip“[124] basiert.

Der einfachste Weg, den neuen Kategorienbaum manuell zu konstruieren ist, die passende Kategorie aus dem Marktstudientitel und der Marktstudienbeschreibung herzuleiten. Man müsste dann jeweils einer Kategorie aus dem Kategorienbaum die Marktstudien anhand des Titels oder der Beschreibung zuweisen. Abhilfe schafft hier die Volltextsuche von MySQL mit der man über mehrere Spalten hinweg in einer Tabelle nach Wörtern oder Sätzen suchen kann. Eine boolsche Volltextsuche in MySQL ist ebenfalls möglich.

[122] Einige von diesen Marktstudien werden später in der Implementierung als Testdaten verwendet
[123] s. S. 77 K. 6, S. 98 K. 7.2 und S. 107 K. 8.1
[124] s. S. 52 K. 4.4

Für eine weitere Vertiefung des Themas MySQL Volltextsuche sei an dieser Stelle auf die Onlinedokumentation von MySQL hingewiesen[125]. Folgendes Beispiel zeigt, wie man mit einem relativ einfachen SQL-Befehl in MySQL eine Volltextsuche vornehmen kann.

```
mysql> SELECT products_title, products_description FROM products_description
 -> WHERE MATCH (products_title,products_description) AGAINST ('Apparel');
```

Listing 19: Volltextsuche in MySQL

Die wichtigsten SQL Funktionen hierbei sind *Match()* und *Against()*. *Match()* gibt die Spalten an, in denen der Begriff gesucht werden soll und der Suchbegriff selber wird in *Against()* als Parameter angegeben. In diesem Fall wird nach dem Begriff *Apparel* sowohl im Titel als auch in der Beschreibung gesucht. Bei einer erfolgreichen Suche wird die Kategorie übernommen. Mit dieser Methode schafft man die Grundlage für einen neuen abgespeckten Kategorienbaum.

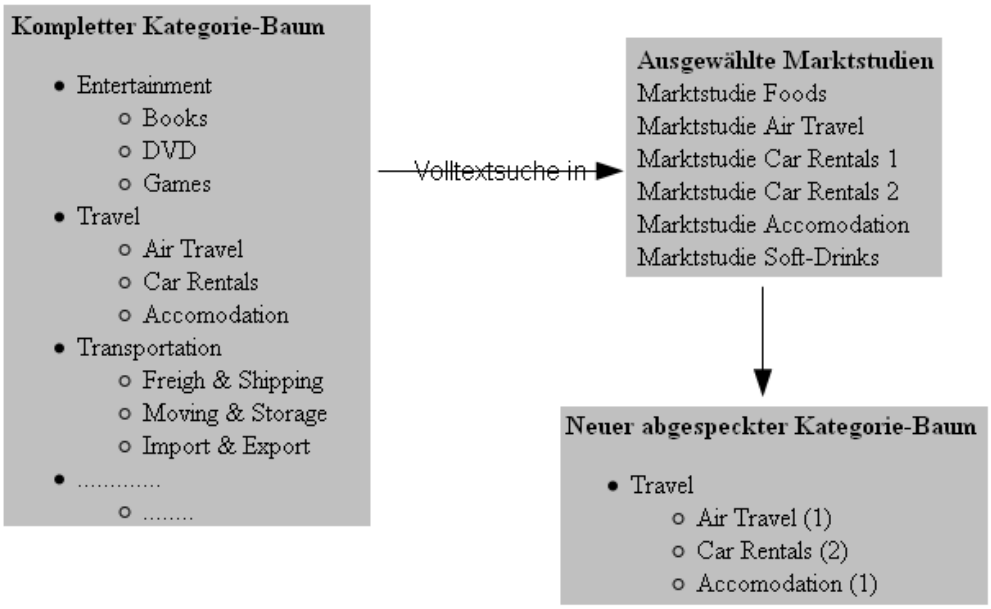


Abbildung 4.4: Graphische Darstellung der Datenbereinigung

Der Kategorienbaum wird am Anfang zunächst ganz klein gehalten und mit der Zeit mit weiteren Marktstudien gefüllt. Auf dieser Basis erfolgt dann die Erstellung einer Ontologie für die Kategorien. Aus praktischer Sicht hat es Vorteile, wenn man sich bei der Ontologie Entwicklung zunächst auf einen anfänglich kleinen Kategorienbaum bezieht, da man dadurch einen besseren Überblick erhält.

[125] http://dev.mysql.com/doc/refman/5.1/de/fulltext-search.html

4.7. Die Ontologie zum neuen Kategorienbaum

Der manuell erstellte und bereinigte Kategorienbaum dient als Basis für die Entwicklung der Kategorienontologie. Bei der Erstellung der Ontologie verwendet man die in Kapitel 4.3 vorgestellte Anwendung *Protege*, um sich einen klaren und visuellen Überblick über alle Kategorien und Ihre Beziehungen untereinander zu verschaffen. Darüber hinaus wird durch *Protege* auch die dazugehörige OWL Datei mitsamt OWL Code automatisch generiert[126]. Der Vorgang zur Entwicklung der Ontologie verläuft iterativ[127], d. h. man beginnt mit einem kleinen Teil des Kategorienbaums, der stufenweise erweitert wird und dadurch an Umfang zunimmt. Folgende Abbildung zeigt den Ausgangspunkt der Entwicklung:

Abbildung 4.5: Auszug aus dem neuen Kategorienbaum

Aus dieser Vorlage wird unter Zuhilfenahme von *Protege* die Ontologie entwickelt und abgebildet. Überträgt man diese Ausgangsdaten in *Protege*, so sieht die Ontologie folgendermaßen aus:

[126] Den kompletten OWL Code zur Kategorienontologie finden Sie auf der beigelegten CD
[127] vgl. S. 53 K. 4.5

Abbildung 4.6: OWL-Darstellung in Protege als Baumansicht

Protege führt automatisch die Klasse bzw. die Oberklasse “Thing“[128] ein, darunter befinden sich alle selbstdefinierten Klassen. Die baumartige Struktur und die Beziehungen der Kategorien untereinander werden in OWL über das Sprachkonstrukt *subClassOf* realisiert. Über *subClassOf* ist bereits der erste Schritt zur Inferenz getan, da *subClassOf* eine transitive Beziehung zwischen den Kategorien definiert[129].

```
<rdf:RDF xmlns="http://localhost/xampp/ontologies/categories_onto.owl#"
 xml:base="http://localhost/xampp/ontologies/categories_onto.owl"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
.....
<owl:Class rdf:about="http://localhost/xampp/ontologies/market_categories#Consumer_Goods"
    />
<owl:Class rdf:about="http://localhost/xampp/ontologies/market_categories#
    Travel_and_Leisure">
 <rdfs:subClassOf>
<owl:Class
 rdf:about="http://localhost/xampp/ontologies/market_categories#Consumer_Goods"
```

[128] s. S. 41 K. 3.3.2.2
[129] vgl. S. 42 K. 3.3.2.3

```
10 /> </rdfs:subClassOf> </owl:Class>
11 <owl:Class rdf:about="http://localhost/xampp/ontologies/market_categories#Travel_Services
       ">
12  <rdfs:subClassOf>
13 <owl:Class rdf:about="http://localhost/xampp/ontologies/market_categories#
       Travel_and_Leisure" />
14  </rdfs:subClassOf>
15 </owl:Class>
16 <owl:Class rdf:about="http://localhost/xampp/ontologies/market_categories#
       Corporate_Travel">
17  <rdfs:subClassOf>
18 <owl:Class rdf:about="http://localhost/xampp/ontologies/market_categories#Travel_Services
       " />
19  </rdfs:subClassOf>
20 </owl:Class>
```

Listing 20: Von Protege generierter OWL Code

Im nächsten Schritt stellt sich die Frage, welche weiteren Verfeinerungen vorgenommen werden können. Grundsätzlich dient die Ontologie zu den Kategorien lediglich als Bezugsontologie für Marktstudien. Dies bedeutet, dass sich die Komplexität der Bezugsontologie in Grenzen hält. Die Definition der Unterklassenbeziehungen mittels *subClassOf* und ggf. der Verwendung von *equivalentClass* reicht bereits vollkommen aus, um aus Sicht der Kategorien Inferenzen abzuleiten.

4.8. Konzeptionieren einer Marktstudienontologie

Eine Marktstudie ist aus Sicht der Ontologie nichts anders als ein Konzept bzw. eine Klasse, die ähnlich wie beim OOP[130] Eigenschaften besitzt. Demzufolge besteht eine Marktstudie aus folgenden Komponenten bzw. Eigenschaften:

- Marktstudientitel
- Jahrgang
- Herausgeber
- Sprachversion
- Allgemeine Beschreibung
- Inhaltsverzeichnis
- Tabellenverzeichnis

[130] Objekt Orientierte Programmierung

- Abbildungsverzeichnis
- Format (PDF, Buch)
- Preis
- Die Webseite, auf der die Marktstudie veröffentlicht werden soll
- Kategorienzugehörigkeit

Zur besseren Visualisierung der Marktstudien mitsamt ihrer Eigenschaften und Beziehungen zu den Kategorien eignen sich Graphen. OWL Dokumente sind ebenfalls gültige RDF Dokumente und das Datenschema von RDF basiert auf Graphen, mit denen man *Statements* veranschaulichen kann[131].

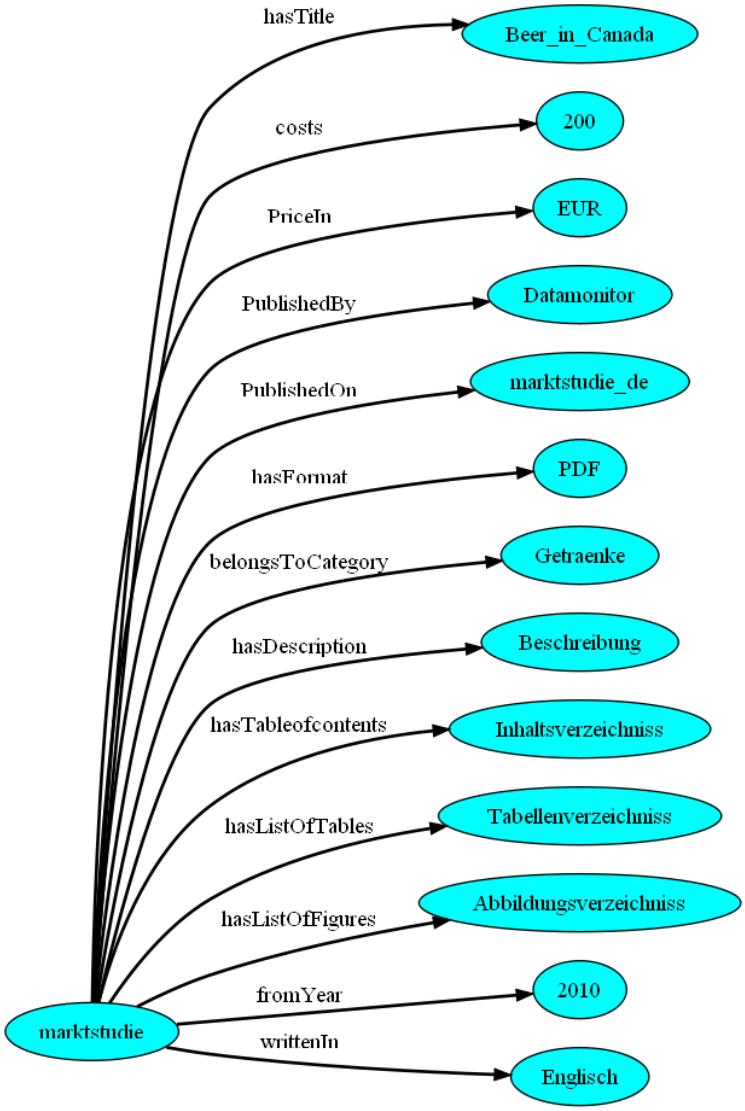


Abbildung 4.7: Skizze des Graphen der Marktstudienontologie

[131] vgl. S. 32 K. 3.2.1

Der Graph veranschaulicht folgende Aussage:

> „Die Marktstudie mit dem Titel „Beer in Canada“ von Datamonitor kostet 200 Euro und gehört zur Kategorie „Getränke“. Die Marktstudie hat das Format PDF und wird auf www.markt-studie.de veröffentlicht“.

Im Gegensatz zu den Kategorien besitzt die Marktstudienontologie mehrere Subjekte bzw. Objekte und beinhalten darüber hinaus Prädikate. Zu den Prädikaten können abstrakte, konkrete *Rollen* und ggf. weitere Rolleneinschränkungen definiert werden[132].
Bei der späteren Implementierung müssen folgende *Rollen* beachtet und definiert werden:

1. **Konkrete *Rollen* (DatatypeProperty)**:
   a) *hasTitle* Auf dieses Prädikat muss ein *String* folgen bzw. ein Datenwert vom Typ *xsd:string*
   b) *costs* Auf dieses Prädikat folgt ein *float* bzw. ein Datenwert vom Typ *xsd:float*
   c) *priceIn* Auf dieses Prädikat folgt ein *String*, also beispielsweise „USD“ oder „EURO“
   d) *hasFormat* Auf dieses Prädikat folgt ein *String* bzw. ein Datenwert vom Typ *xsd:string*
   e) *hasDescription*, *hasTableOfContents*, *hasListOfTables*, *hasListOfFigures* Auf diese Prädikate folgt ein String bzw. ein Datenwert vom Typ *xsd:string*
   f) *fromYear* Das Standardformat für den Jahrgang ist Monat/Jahr, demzufolge eignet sich dafür der Datentyp *xsd:gYearMonth*
   g) *writtenIn* Marktstudien werden in verschiedenen Sprachen verfasst. Über dieses Prädikat folgt die Sprachangabe. Dafür verwendet man ebenfalls den Datentyp *xsd:string*
2. **Abstrakte *Rollen* (ObjectProperty)**:
   a) *belongsToCategory* Bei der späteren Implementierung spielt dieses Prädikat eine große Rolle. Es ist hauptverantwortlich für die Zuordnung der Marktstudie zu einer oder mehreren Kategorien.
   b) *publishedOn* Generell muss redaktionell entschieden werden, auf welchem/welchen der drei Marktstudienportale die Marktstudie veröffentlicht werden soll. Daraus folgt, dass auf dieses Prädikat entweder der Wert „www.markt-studie.de“, „www.reports-research.com“ und/oder „www.estudio-mercado.es“ folgt. Dafür

[132] vgl. S. 41 K. 3.3.2.2, List. 11

eignen sich am besten OWL Klassen, die für jedes Portal eigens definiert werden.

c) *publishedBy* Sowie wie Bücher haben auch Marktstudien einen Herausgeber. Die Autoren der Marktstudien arbeiten zumeist für einen Herausgeber und demzufolge sind Herausgeber und Autoren äquivalent. Ferner muss beachtet werden, dass nur registrierte Partner der dytec GmbH Marktstudien im Portal veröffentlichen dürfen. Demzufolge muss zusätzlich eine „Publisherontologie“ definiert werden[133].

Zusammenfassend betrachtet benötigt man später insgesamt drei Ontologien: Die Kategorienontologie, die Ontologie zu den Marktstudien Hersteller[134] und zuletzt die Marktstudienontologie. Letztere greift über die definierten *Namespaces* auf die ersten beiden Ontologien zu.

4.9. Ausblick auf die spätere Implementierung

In der späteren Realisierung muss beachtet werden, dass zahlreiche Marktstudien nicht nur einer sondern mehreren Kategorien zugeordnet werden. Ein weiterer wichtiger Aspekt ist die Bildung von Marktstudien Individuen, also die Instanziierung von konkreten Objekten bzw. Individuen aus der Marktstudien Klasse. Hierbei stellt sich die Frage, ob die Kategorienzuordnung auf Basis von Individuen oder Klassen geschehen soll. Ein großer Fehler wäre es, die Marktstudien dem Typ „Kategorie“ zuzuordnen, z. B. :

```
1 <rdf:Description rdf:about="MarktstudiePullover">
2  <rdf:type rdf:resource="Apparel">
3 </rdf:Description>
```

Listing 21: Falsche Deklaration eines Marktstudien Individuums

Der Fehler besteht hier in der Annahme, dass das Individuum “MarktStudiePullover“ vom Typ „Apparel“ ist. Dies ist logisch falsch, da ja generell Marktstudien von der Klasse “Marktstudie“[135] abgeleitet werden. Die Basis für so eine Marktstudien Klasse stellt die im Kapitel 4.8 konzipierte Ontologie dar und über das Prädikat „belongsToCategory“ erfolgt die Verbindung einer Marktstudie mit einer oder mehreren Kategorien aus der Kategorienontologie. Die Zuordnung von Marktstudien zu mehreren Kategorien kann unter Zuhilfenahme von logischen Konstruktoren realisiert werden[136]. Die einfachste Variante ist

[133] s. Anhang B, S. 124, List. 34
[134] Publisherontologie
[135] s. Anhang B, S. 124, List. 32
[136] s. S. 44 K. 3.3.2.4

jedoch die Verwendung des OWL Sprachkonstrukts *owl:someValuesFrom* mit dem man ausdrücken kann, dass die Klasse “Marktstudie“ mindestens einer Kategorie angehören muss[137].

Mit Hilfe von Ontologien wird die Möglichkeit geschaffen, Informationen, in diesem Fall Marktstudien mitsamt ihrer Kategorien einer semantischen Bedeutung zuzuordnen bzw. sie semantisch auszuzeichnen. Um dies zu erreichen, müssen die Texte bzw. Wörter semantisch markiert werden. Hierbei stellt sich die Frage, wie diese Auszeichnungen in den Text gelangen. Vor der Überführung in OWL muss zunächst eine *Wissensaquisition*[138] in Form einer *Text Extraction* und ggf. eines *Text Mining* stattfinden. Über OWL wird das Wissen zwar in einer bestimmten formalen Art und Weise repräsentiert und hinterlegt, doch vorher muss das Wissen selbst in einer geeigneten Form bereitgestellt werden[139]. Das durch *Text Extraction* und *Text Mining* extrahierte Wissen dient als Grundlage für die Erstellung einer Ontologie bzw. einer OWL Datei.

[137] s. S. 110 K. 9.1.1, List. 30
[138] [HQW08], S. 17
[139] ebenda, S. 17

5. Text Extraction

Dieses Kapitel befasst sich mit dem Begriff *Text Extraction* und erläutert, welcher technisch-wissenschaftlichen Disziplin er zuzuordnen ist. Desweiteren wird beleuchtet, welche Ziele und Zielsetzungen das Fachgebiet *Text Extraction* verfolgt. Außerdem werden die Extraktionsmethoden von Luhn und Edmundson vorgestellt. Darüber hinaus wird das Thema *Text Mining* aufgegriffen und erklärt, in welcher Beziehung dieses Fachgebiet zur *Text Extraction* steht. Hier stellt sich v. a. die Frage, wie nah aneinander die Grenzen zwischen *Text Extraction* und *Text Mining* verlaufen.

5.1. Begriffserklärung und Ursprung des Themas

Zunächst muss darauf hingewiesen werden, dass *Text Extraction* äquivalent ist zu dem Fachgebiet *Automatic Summarization*[140]. In vielen englischsprachigen Fachbüchern, die dieses Thema behandeln, bevorzugt man den Begriff *Automatic Summarization*. Diese Diplomarbeit setzt beide Begriffe gleich, d. h. *Text Extraction* meint *Automatic Summarization*.

Das Fachgebiet der *Text Extraction* bzw. *Automatic Summarization* ist höchst interdisziplinär[141] [142], das Thema vereinigt unterschiedliche Fachgebiete und Themenbereiche wie Psychologie, Information Retrieval, Linguistik, künstliche Intelligenz sowie Sprachverarbeitung, Bibliothekswissenschaft und Statistik[143] [144]. Zusammenfassend betrachtet liegt der Ursprung von *Text Extraction* in der kognitiven Wissenschaft und im *Natural Language Processing* (NLP). Aus Gründen der Komplexität geht diese Diplomarbeit nicht auf alle o. g. Themenbereiche ein, Aspekte der theoretischen und technischen Umsetzung der *Text Extraction* stehen stärker im Vordergrund als beispielsweise die Erläuterung der kognitven Wissenschaft.

[140] Bei der Recherche zu *Text Extraction* im Internet oder auch in Bibliotheken wird regelmäßig auf Literatur zu *Automatic Summarization* verwiesen
[141] fachgebietsübergreifend
[142] [Man98], S. 21
[143] ebenda, S. 21
[144] [EN98], S. 1

5.2. Einführung: Bedeutung und Zielsetzungen

Die *Text Extraction* beschreibt Methoden zur automatischen Zusammenfassung von Texten[145]. Ziel ist es, einen großen Satz an Informationen zu reduzieren, so dass lediglich einige relevante Aspekte herausgefiltert werden[146]. Die extrahierten, gekürzten Informationen repräsentieren den relevanten Inhalt eines Textes und müssen in einer Art und Weise bereitgestellt bzw. präsentiert werden, die den Bedürfnissen des Benutzers genügt und aus der er einen Nutzen ziehen kann[147]. Inderjeet Mani, Autor und Experte auf dem Gebiet der *Text Extraction*, bezeichnet dies in seinem Lehrbuch „Automatic Summarization“[148] als „condensed information“[149] [150]. Programme zur automatischen Zusammenfassung müssen darauf ausgerichtet sein, die zusammengefassten Informationen für den Menschen so weit zu präsentieren, dass sie aus menschlicher Sicht konsumierbar sind[151].

Bei der *Text Extraction* spielt die Beziehung zwischen der Zusammenfassung und deren Eingabe[152] eine bedeutende Rolle. Man unterscheidet zwei Arten von Zusammenfassungen: *Extract* und *Abstract*. Ein *Extract* ist eine Zusammenfassung, die eine Liste aller Schlüsselwörter[153] [154] aus der Eingabe enthält, d. h. bei einem *Extract* werden ausnahmslos alle vorhandenen Wörter aus der Eingabe übernommen und in einer Liste hinterlegt. Im Gegensatz dazu beinhaltet ein *Abstract* nur die relevanten Sätze und Abschnitte. Somit stellt ein *Abstract* einen Abriss der Originalinformation dar, die den Kerngegenstand des Themas widerspiegelt. Demzufolge bietet ein *Abstract* einen höheren Informationsgehalt als der *Extract*, da letzterer zu umfangreiche und zu wenig präzise Informationen enthält. Allein der Aspekt des hohen Informationsgehalts eines *Abstract* stellt bei der späteren Implementierung einen Vorteil dar: Die Zusammenfassung kann auf Basis eines *Abstract* erstellt werden. Eine weitere Möglichkeit ist eine Kombination beider Ansätze, jedoch muss der *Extract* Vorgang ein wenig modifiziert werden: Die Erzeugung einer Liste ist zwar immer noch das primäre Ziel, jedoch werden nur diejenigen Wörter in die Schlüsselwortliste aufgenommen, die wirklich relevant[155] sind. Im Anschluss daran wird auf Basis dieser Liste ein *Abstract* erzeugt.

[145] daher *Automatic Summarization*
[146] [EN98], S. 98
[147] [Man98], S. 1
[148] s. [Man98]
[149] ebenda, S. 1
[150] engl.für komprimierte Information
[151] [EN98], S. 299
[152] engl.: input
[153] Begriffe, Nomen, Nominal Phrasen
[154] [Man98], S. 6
[155] s. Luhn Algorithmus in K. 5.4.1

Neben dem reinen Text als Eingabemedium für die Zusammenfassung sind mittlerweile auch andere Eingabemedien möglich, so z. B. multipele bzw. einzelne Textdokumente oder auch XML.

Der theoretische und praktische Grundstein der automatischen Zusammenfassung wurde bereits in den späten fünfziger Jahren gelegt. Einer der Pioniere auf diesem Gebiet war der deutsche Informatiker H. P. Luhn. Luhn stellte seine Methode zur automatischen Textabstraktion im IBM Journal Ausgabe April 1958 vor[156]. Seine theoretischen Arbeiten zur Informationsabstraktion werden in einem späteren Kapitel näher erläutert[157].

5.2.1. Wozu Text Extraction und für wen?

Recherchen im WWW sind heutzutage meist relativ zeitintensiv, da das WWW aufgrund seines ständigen Wachstums seit den Anfangsjahren von Informationen überflutet wird. Es erweist sich als ausgesprochen schwierig, aus diesen zahlreichen Informationen zügig die wichtigsten auszuwählen und manuell zusammenzufassen. Im wirtschaftlichen Bereich, v. a. im geschäftsorientierten Prozess, ist es aber von enormer Wichtigkeit, schnell an Informationen zu gelangen und diese zusammenzufassen, da auf Basis dieser Informationen kritische Entscheidungen getroffen werden müssen. Die Technologie der *Text Extraction* kann hierbei Abhilfe schaffen – mit ihrer Hilfe ist es möglich, ausschließlich die relevanten Informationen zusammenzufassen. Demzufolge muss ein *Text Extraction* Programm noch eine weitere elementare Aufgabe erfüllen: „Text Summarization is the process of distilling the most important information from a source (or sources) to produce an abriged version for a particular user [. . .].“[158] Laut dieser Aussage ist das Produkt bzw. das Endergebnis einer *Text Extraction* i. d. R. auf einen bestimmten Benutzer ausgerichtet. Falls der Benutzer aus dem medizinischen Bereich kommt, muss bei der Zusammenfassung auf medizinische Fachtermini geachtet werden. Das Prorgramm zur *Text Extraction* muss demzufolge einem bestimmten Benutzerprofil angepasst werden[159].

5.3. Text Mining – eine Alternative?

Text Mining betreibt diverse Entdeckungsverfahren[160] bzw. eine Reihe von statistischen Methoden, um Anomalien oder andere „interessante“ neue Informationen in einem sehr

[156] s. [Luh58]
[157] s. S. 68 K. 5.4.1
[158] [Man98], S. 1
[159] vgl. S. 66 K. 5.2
[160] [Man98], S. 4

umfangreichen Text zu finden. Texte stellen i. d. R. unstrukturierte Daten dar, mit Hilfe von *Text Mining* und seinen dazugehörigen Werkzeugen, können aus digital vorliegenden Texten neue relevante, sachliche und inhaltliche Zusammenhänge extrahiert werden[161]. Der Fokus von *Text Mining* liegt nicht auf dem Zusammenfassen von Informationen, sondern eher auf der Charakterisierung von „Singularitäten“[162]. Generell ist das Thema *Text Mining* sehr breit gefächert. Eine große Rolle spielt *Text Mining* in der späteren Implementierung bei der Berechnung von Worthäufigkeiten in einem Text, die essenziell für die Erstellung eines *Abstract* ist. Ziel ist es, aus einer Vielzahl an *Text Mining* Methoden eine geeignete statistische Methode auszuwählen, um die *Text Extraction* zu unterstützen.

5.4. Die klassischen Extraktionsmethoden

5.4.1. Extraktionsmethode nach Luhn

Der deutsche Informatiker H. P. Luhn war einer der ersten, der eine Methode zur Erstellung eines *Textabstract* entwickelte. Seine Arbeit veröffentlichte er in der Ausgabe 1958 des IBM Journals unter dem Titel „The Automatic Creation of Literatur“[163]. Darin beschreibt Luhn den Algorithmus zur Satzextraktion. Sein Ziel war es, aus einem Zeitungsartikel das zentrale Thema, den Hauptgegenstand, des Artikels zu ermitteln. Luhns Algorithmus ist satzorientiert, d. h. bei der Erstellung eines *Abstract* werden nur die Sätze ausgewählt, die eine hohe Relevanz besitzen und Kernpunkte des Textes widerspiegeln. Um zu ermitteln, welche Sätze eine hohe Signifikanz besitzen, damit *Abstracts* erfolgreich erstellt werden können, muss eine Maßnahme erfolgen, die den Informationsgehalt aller Sätze benotet. Auf Basis der Analyse der einzelnen Wörter im Text leitet man den Signifikanzfaktor ab. Laut Luhn hängt die Signifikanz eines Satzes von folgenden Faktoren ab:

- Die Häufigkeit, in der ein relevantes Wort vorkommt
- Die Position von relevanten Wörtern in einem Satz

Nach Luhns Meinung basiert der erste Faktor auf der Tatsache, dass Autoren normalerweise bestimmte Wörter bzw. Begriffe wiederholen um ihre Argumente zu stützen und dadurch das eigentliche Thema in den Vordergrund zu stellen: „[. . .] a writer normally repeats certain words as he advances or varies his arguments and as he elaborates on an aspect of a subject.“[164]

[161] [HQW08], S. 1
[162] [Man98], S. 4
[163] s. [Luh58]
[164] [Luh58], S. 160

Beim zweiten Faktor stützt sich Luhn auf seine empirische Analyse von Wortpositionen innerhalb eines Satzes. Nach seiner Erfahrung ist ein Satz nur dann relevant, wenn zwischen zwei relevanten Wörtern nicht mehr als vier oder fünf nicht-relevante Wörter stehen.

Prinzipiell muss man bei der Implementierung zunächst eine Art Inventarliste erstellen, die alle vorkommenden Wörter mitsamt ihrer Häufigkeit enthält. Luhn schlägt vor, bzgl. der Häufigkeit eine obere und untere Grenze zu bestimmen, mit der man zu selten und zu häufig vorkommende Wörter ausschliesst. Desweiteren vermutet Luhn, dass die wirklich relevanten bzw. signifikanten Wörter im mittleren Häufigkeitsbereich liegen, da weder Begriffe mit hoher noch mit niedriger Häufigkeit ein signifikantes Wort darstellen. Um einen optimalen Grenzwert zu bestimmen, muss man auf Erfahrungswerte zurückgreifen. Um bei der Implementierung die Signifikanz eines Satzes zu errechnen, stellt Luhn auf Basis der beiden vorhin genannten Faktoren folgende Formel auf:

Es sei:
SignificanceWords = Anzahl der signifikanten Wörter innerhalb eines Satzes
AllWords = Anzahl der Wörter innerhalb eines Satzes
Sigfactor = Signifikanzfaktor des Satzes

$$Sigfactor = \frac{(SignificanceWords)^2}{AllWords} \tag{13}$$

Luhns Methode ist satzorientiert, man nimmt einen Satz und vergleicht jedes Wort mit den in der Inventarliste vorkommenden Wörtern. Zusammenfassend betrachtet werden bei Luhns Methode folgende Schritte vorgenommen:

1. Erstelle aus dem Text eine Inventarliste mit allen signifikanten Wörtern.
2. Gehe jetzt satzweise durch.
3. Kalkuliere zunächst die Anzahl aller Wörter inklusive Stoppwörter[165] (auch: Funktionswörter) in einem Satz.
4. Nehme ein Wort aus einem Satz und vergleiche es mit der Inventarliste. Falls es zu einer Übereinstimmung kommt, Wortanzahl addieren.
5. Nehme nächstes Wort und wiederhole Schritt 4, fahre auf diese Weise fort, bis das Satzende erreicht ist.

[165] bsp.: bestimmter Artikel oder unbestimmter Artikel

6. Am Satzende die o. g. Formel anwenden, um die Signifikanz des Satzes zu ermitteln, danach Satz speichern.
7. Mit nächstem Satz fortfahren und Schritte 3 bis 6 wiederholen, bis das Textende erreicht ist.
8. Am Textende dann alle gespeicherten Sätze ausgeben.

Seine Methode wandte Luhn an 50 Zeitungsartikeln mit jeweils unterschiedlicher Anzahl von Wörtern an. Die Wortanzahl der ausgewählten Artikel betrug zwischen 350 bis 4500. Aus diesen Artikeln wurden mit Erfolg sinngemäße Zusammenfassungen erstellt. Im Großen und Ganzen bestand die Zusammenfassung aus automatisch ausgewählten in der richtigen Reihenfolge ausgegebenen relevanten Sätzen. Luhn zeigte damit, dass sein Algorithmus tatsächlich in der Lage ist, *Abstracts* zu erstellen. Zwar besteht der *Abstract* nur aus einer aneinanderreihung von Sätzen, dies sind aber die relevantesten des gesamten Textes.

Die nach Luhns Algorithmus erstellten Zusammenfassungen sind zuvelässig, konsistent und stabil[166]. Doch dieser Algorithmus birgt einige entscheidende Nachteile: so basiert Luhns Algorithmus lediglich auf seiner Erfahrung mit Zeitungsartikeln. Es kann vorkommen, dass ein Autor einen individuellen Schreibstil bevorzugt, der vom allgemeinen abweicht. Die von Luhn erstellte Formel zur Berechnung der Satzsignifikanz würde dann ggf. nicht mehr korrekt greifen. Zudem nannte Luhn keine Methode, wie man die von ihm benannte mittlere Häufigkeit für eine Inventarliste berechnet.

Bzgl. des erstgenannten Nachteils ist Luhn der Ansicht, dass seine Methode korrigiert und verbessert werden kann. So könne z. B. die *Summarizer* Anwendung so implementiert werden, dass sie nur auf bestimmte Themengebiete oder auch Untersuchungsgebiete zugeschnitten wird[167] [168].

Die zweite Schwachstelle in Luhns Methodik kann man ggf. durch statistische Methoden aus dem *Text Mining* oder durch andere Extraktionsmethoden beheben[169]. Auf diese Weise kann später in der Implementierung Luhns Algorihmus als erster Ansatz angewendet werden, um aus einer Marktstudienbeschreibung ein *Abstract* zu erstellen.

[166] [Luh58], S. 168
[167] vgl. S. 67 K. 5.2.1
[168] s. Edmundsons Methode in K. 5.4.2
[169] ebenda

5.4.2. Extraktionsmethode nach Edmundson

Auf Grundlage von Luhns Arbeit entwickelte Edmundson rund zehn Jahre später eine neue Methode zur Textabstraktion. Seine Arbeit wurde unter dem Titel „New Methods in Automatic Extraction“[170] veröffentlicht.

Wie Luhn war auch Edmundson der Ansicht, dass die richtige Auswahl signifikanter Sätze zu einer erfolgreichen Zusammenfassung führt. Allerdings empfand Edmundson die Methode von Luhn nicht mehr als eine reine *Abstract* Methode, sondern eher als ein *Automatic extract*[171]. Nach Edmundson sollen bei der Auswahl der signifikanten Sätze folgende zusätzliche Faktoren beachtet werden:

- *Cue words*[172]
- *Title words*[173]
- *Heading words*[174]
- *Sentence location*[175]

Im Gegensatz zu Luhn benutzte Edmundson bei der Entwicklung seiner Methode keine Zeitungsartikel, sondern naturwissenschaftliche Dokumente z. B. aus dem Fach Chemie. Dokumente aus dem Fach Chemie haben i. d. R. eine bestimmte Struktur: Sie beginnen mit einer allgemeinen Zusammenfassung zur Problemstellung, es folgt ein Abschnitt zur Orientierung, anschließend ein Abschnitt über die Methoden, danach ein Diskussionsabschnitt und zuletzt die Schlussfolgerung zum behandelten Problem. Edmundsons Methode basiert demzufolge auf strukturierten Dokumenten, weshalb aus Edmundsons Sicht die Faktoren *Title Words*, *Heading Words* und *Sentence Location* von großer Wichtigkeit sind. Eine andere bedeutende Rolle bei Edmundsons Methode spielt das sogenannte *Cue Dictionary*[176]. Daher stellen die *Cue Words* auch eine wichtige Komponente bei der Erstellung eines *Abstract* dar. Generell ist die Erstellung eines Dictionary und eines Glossars wichtig und elementar. Dictionaries und Glossare sind Wortlisten, die sich darin unterscheiden, dass ein Dictionary allgemeine Wörter enthält und unabhängig von dem zu abstrahierenden Textdokument ist. Im Gegensatz dazu bezieht sich ein Glossar immer auf ein bestimmtes Dokument und beinhaltet demzufolge dessen Wörter.

[170] s. [Edm68]
[171] [Edm68], S. 23
[172] Stichwörter oder auch Signalwörter
[173] Wörter aus dem Titel
[174] Wörter aus Überschriften eines Textabschnittes
[175] Der Standort eines Satzes
[176] Stichwort Wörterbuch oder auch Wort-Liste

Auf Grundlage der vorhin genannten Faktoren und Komponenten wie Dictionaries und Glossare schlug Edmundson vier grundlegende Methoden für eine erfolgreiche Textzusammenfassung vor:

- *Cue Method*[177]
- *Key Method*[178]
- *Title Method*[179]
- *Location Method*[180]

Cue Method

Im Prinzip geht es bei dieser Methode um die Erstellung eines Dictionary, das Wörter aus ausgewählten Textdokumenten beinhaltet[181][182]. Die Maschine wird demnach vorerst mit Informationen gefüttert. Dabei ist zu beachten, dass diese Textdokumente fachspezifisch sind. Mit der bloßen Auflistung aller Wörter im Dictionary ist die Arbeit noch nicht getan, zusätzlich erfolgt eine Klassifizierung der Wörter, die wie folgt unterteilt sind:

- *Bonuswörter* – Dies sind Wörter, die fachspezifisch sind
- *Stigmawörter* – Dies sind Wörter, die keine Relevanz in Bezug auf ein Fachgebiet besitzen
- *Nullwörter* – Dies sind Wörter, die zu oft vorkommen, z. B. Stoppwörter

Auf Grundlage dieser Klassifizierung wird die Signifikanz eines Satzes errechnet. Die *Cue Method* ahmt die menschliche Methode der Zusammenfassung nach. Ein Mensch fasst einen Text auf Grundlage seiner Erfahrung und seines Wissen zusammen. Bei der *Cue Method* stellt das *Cue Dictionary* das gesamte Wissen und die Erfahrung einer Maschine dar.

Key Method

Die *Key Method* ist äquivalent zu Luhns Methode, man sammelt alle Wörter eines Textdokuments und hinterlegt sie anstatt in einem Dictionary in einem Glossar, da Glossare ja immer in einem Bezug zu einem Textdokument stehen. Das Glossar wird dann für die Berechnung der Satzsignifikanz verwendet[183].

[177] Stichwort Methode
[178] Schlüsselwort Methode
[179] Titel Methode
[180] Methode zur Bestimmung eines Satz Standortes
[181] Damals verwendete Edmundson 100 Textdokumente aus dem Gebiet der Chemie
[182] [Edm68], S. 30
[183] vgl. mit Luhns Methode

Title Method

Diese Methode verfolgt das gleiche Ziel wie die *Key Method*, allerdings mit dem Unterschied, dass nur Wörter in einem Glossar aufgenommen werden, die in einem Titel, d. h. in dem Dokumententitel oder dem Titel eines Abschnitts[184], auftreten. Die *Title Method* basiert auf der Annahme, dass Autoren durch einen Titel im Großen und Ganzen die Hauptthematik eines Textes bzw. Abschnitts erfassen können. Demzufolge erhalten Wörter aus einem Titel eine höhere Gewichtung und folglich enthält das Glossar bei der *Title Method* keine *Nullwörter*. Allerdings ist diese Methode nicht frei von Problemen: z. B. übertrifft gelegentlich die Gewichtung eines Titels eines Abschnitts die Gewichtung des Textdokumententitels oder des übergeordneten Abschnitstitels, so dass die richtige Reihenfolge der relevanten Sätze in der Zusammenfassung nicht mehr gewährleistet ist. Demnach muss die Gewichtung noch einmal verfeinert werden: Die Gewichtung eines Textdokumentenitels ist höher einzustufen als der Titel eines Abschnitts[185].

Location Method

Diese Methode beruht auf der Annahme, dass ein bestimmter Satz unter einer Überschrift oder einem Titel eine höhere Relevanz besitzt. Sätze, die die Kernthematik beinhhalten und ebenfalls als relevant einzustufen sind, erscheinen i. d. R. entweder am Anfang eines Textes bzw. Abschnitts oder gegen Ende. Die hierbei mittels der *Location Method* ermittelten Wörter werden ebenfalls in einem Dictionary gesammelt. Die Berechnung der endgültigen *Standort Gewichtung* ergibt sich aus der Gewichtung der Überschrift, unter der der Satz steht, addiert mit der Gewichtung des Satzes.

Edmundson testete damals mit 100 fachspezifischen Dokumenten[186] [187], von jeweils 3000 bis 4000 Wörtern die einzelnen Methoden und fand heraus, dass die Key-Method nicht effizient arbeitet, da sie im Gegensatz zu den anderen Methoden die wenigsten Sätze auswählte, obwohl alle Methoden dieselben Textdokumente abstrahierten[188]. Daher empfahl Edmundson entweder die *Cue Method* oder die Kombination *Cue-Title-Location Method* zu verwenden. Edmundson bevorzugte die *Cue Method*, die er am Ende seines Papers als *Cue Dictionary Programme*[189] ausführlich vorstellte. Bei der Erstellung einer Zusammenfassug mit der *Cue Method* müssen folgende Schritte vorgenommen werden[190]:

[184] Überschrift eines Abschnitts
[185] [Edm68], S. 31
[186] Edmundson verwendete vornehmlich Textdokumente aus dem Fachgebit Chemie
[187] [Edm68], S. 32
[188] ebenda, S. 33
[189] ebd., S. 33
[190] ebd., S. 33

1. Vergleiche jedes Wort eines Textdokuments mit den Wörtern aus dem *Cue Dictionary*.
2. Ermittle dabei die Bonus-, Stigma- und Nullwörter. Alle Bonuswörter werden mit b > 0 gewichtet, alle Stigmawörter mit s < 0 und alle Nullwörter mit n = 0.
3. Ermittle den *Cue Weight*[191] pro Satz durch Addition der Wortgewichtungen von b, s und n[192].
4. Liste alle Sätze nach ihrer Gewichtung in absteigender Reihenfolge auf.
5. Wähle diejenigen Sätze aus, deren Gewichtung über einem bestimmten Schwellenwert liegt, z. B. sollen nur die Sätze ausgewählt werden, die eine positive Gewichtung besitzen.
6. Wähle alle Überschriften aus.
7. Vereinige die ausgewählten Sätze unter einem korrekten Titel.
8. Ausgabe des Titels, Autors und der Ergebnisse aus Schritt 7.

Ähnlich wie Luhn bei seiner Methode liefert Edmundson nur theoretische Ideen, wie eine automatische Zusammenfassung funktionieren soll und kann. Dennoch kann man auf Basis dieser Ideen ein Programm aufbauen. Edmundsons Methode ist im Vergleich zu Luhns Methode komplexer und aufwendiger. Auch ist sie auf naturwissenschaftliche Dokumente spezialisiert und nur bei strukturierten Dokumenten anwendbar. Dieses Problem kann man heutzutage jedoch mit XML, das die Erstellung von semi-strukturellen Dokumente ermöglicht, lösen[193].

Eine Verbesserung von Edmundsons Methode ist ebenfalls möglich, v. a. bei der Verwendung der *Cue Method*. Hierbei kann man mit Hilfe eines Verfahrens aus dem *Text Mining* die Häufigkeit von Wörtern in verschiedenen Texten ermittlen und daraus ein *Cue Dictionary* aufbauen.

5.4.3. Fazit

Mit ihren Ideen legten Luhn und Edmundson den Grundstein für die Entwicklung von *Summarizer* Systemen. Deutlich wird, dass beide verschiedene Ansätze zur Abstractgenerierung verwenden. Während Luhn sich auf statistische Methoden stützt, also Berechnungen von Worthäufigkeiten, verwendet Edmundson das sogenannte Signalwortverfahren.

[191] Stichwort Gewicht pro satz
[192] siehe Punkt 2
[193] s. S. 27 K. 3.1

Auf Basis von Luhn und Edmundson entstanden weitere, moderne Variatonen der Extraktionsmethoden. Allerdings reichen die beiden hier vorgestellten Extraktionsmethoden für die spätere Implementierung aus. Die Methoden von Luhn und Endmundson mögen zwar veraltet sein, die ihnen zugrunde liegenden Ideen sind aber dennoch aufschlussreich und bieten für die spätere Entwicklung gute Ansätze. Weitere Extraktionsmethoden kann man in [Aut08] nachlesen.

5.5. Die Rolle der Text Extraction in der Anwendung

Die *Text Extraction* alleine reicht jedoch nicht für eine Wissensakquise aus. Ihre Aufgabe ist es, einzig und allein die Informationsflut beim Import der Marktstudiendaten zu minimieren. V. a. sollen die Beschreibungen zu den Marktstudien auf das wesentliche reduziert werden. Als Ergebnis erhält man ein oder meherere *Abstracts* der Martkstudienbeschreibung. In den *Abstracts* stehen nur noch die relevantesten Wörter bzw. Begriffe auf deren Grundlage dann die Kategorisierung erfolgt. Schlussendlich findet die Generierung der OWL Datei statt. Zwischen dem *Abstract* und der Generierung der OWL Datei muss jedoch ein Zwischenschritt erfolgen, da die Aufgabe der *Text Extraction* schon bei der Erstellung des *Abstract* endet. Die Marktstudienkategorien sind zwar in Form von OWL vorhanden[194], doch ist diese Datei unantastbar, da sie lediglich als Bezugsontologie bei der Erzeugung der OWL Datei dient. Es fehlt also eine Methode zur Zuordnung der Marktstudien zu den Kategorien in Form von OWL. Da die Verwendung der Kategorien in der OWL Datei tabu ist, müssen die Kategorien zunächst noch in einer bestimmten Form bereitgestellt werden[195]. Dafür bieten die vorhandenen Daten im Marktstudienportal einen guten Ansatzpunkt. Grundidee ist es, mit Hilfe von *Text Mining* eine Analyse aller Marktstudien pro Kategorie vorzunehmen, in der alle Wörter bzw. Begriffe gesammelt werden, die eine mittlere Häufigkeit aufweisen. Die mittels *Text Mining* gefundenen Begriffe werden dann in einer Wortliste hinterlegt, ähnlich wie bei einem nach Edmundsons *Cue Method* erstelltem *Cue Dictionary*[196]. Das Ziel ist demnach, eine Art Dictionary für Kategorien zu erstellen.

Anhand dieses Dictionary findet daraufhin die Kategorisierung statt, die wie folgt ausgeführt wird:

1. Erstelle einen *Abstract*.
2. Nehme ein Wort aus dem *Abstract* und suche es in dem Kategoriendictionary.

[194] s. S. 58 K. 4.7
[195] s. S. 63 K. 4.9
[196] vgl. S. 71 K. 5.4.2

3. Falls das Wort in einem Kategoriendictionary auftaucht, wird die Marktstudie der Kategorie zugeordnet, andernfalls das nächstes Wort nehmen und mit Schritt 2 fortfahren.
4. Generiere die OWL Datei.

Wie man an diesen Schritten erkennen kann, liegt eine gewisse Ähnlichkeit zu Edmundsons *Cue-Method* vor. Auch erfolgt durch diese Methode die mehrmalige Kategorienzuordnung.

Das Thema *Text Mining* rückt somit in den Vordergrund und nimmt in der späteren Implementierung eine zentrale Rolle ein.

5.6. Die Rolle von Text Mining

Text Mining nimmt in der späteren Implementierung eine unterstützende Rolle ein, v. a. in Bezug auf die statistischen Methoden von Luhn[197] und auf Edmundsons *Cue Dictionary*. Da *Text Mining* ein sehr breites Spektrum an Unterthemen umfasst, gilt es nun, die passende Methode zu eruieren. Luhns Inventarliste könnte z. B. unter Zuhilfenahme der Berechnungen der relativen Häufigkeit eines Wortes im Text erstellt werden. Ein anderes statistisches Verfahren ist die Berechnung von Häufigkeitsklassen, mit der man Wörter im Text klassifizieren kann. Welche weiteren adäquaten *Text Mining* Methoden geeignet sind, stellt sich erst während des Prozesses der Implementierung[198] heraus. Die Problemstellung determiniert die zu benutzende Methode.

[197] Erstellung der Inventarliste
[198] s. S. 85 K. 6.3, S. 93 K. 6.4 und S. 98 K. 7

6. Implementierung der Text Extraction

In diesem Kapitel wird auf die Realisierung und auf die Implementierung der *Text Extraction* eingegangen. Zunächst werden jedoch die grundlegenden Schnittstellen und Klassen vorgestellt, die sowohl in der *Text Extraction* als auch beim *Text Mining* ihre Verwendung finden.

Für die Implementierung wurde eine Testumgebung mitsamt Testserver und Testdatenbank aufgesetzt. Einige relevante Tabellen[199] aus der Datenbank des Marktstudienportals wurden in diese Testumgebung kopiert. Auf die Übertragung des kompletten Inhalts der Tabellen wurde aus entwicklungstechnischen Gründen verzichtet, stattdessen nur die Struktur der Tabellen übernommen. Der Vorteil liegt darin, dass man sukzessiv beobachten kann, wie sich statistische Meßdaten mit dem Zuwachs von Texten oder Dokumenten[200] [201] verändern. Dieser Vorgang auch einen praktischen Vorteil: Über die Jahre hinweg wurden Marktstudien in die Datenbank importiert, wobei man in letzter Zeit erkannte, dass etliche Marktstudien fehlerhaft kategorisiert wurden. Würde man die „Originaldaten“ aus den Marktstudien behalten, würde dies später zu fehlerhaften Ergebnissen führen, z. B. Beispielsweise bei der Erstellung der Kategorienwortliste.

6.1. Grundlegende Schnittstellen und Klassen

Insgesamt geht es bei der *Text Extraction* und dem *Text Mining* um Analysen von Text. Den Grundstein dafür stellt die Schnittstelle *ITextArticle*. Die von dieser Klasse realisierte Schnittstelle repräsentiert einen Text bzw. ein Text-Artikel-Objekt. Über diese Schnittstelle bzw. Klasse kann jegliche Textart dargestellt werden. Neben Marktstudienbeschreibungen z. B. News Artikel und Online Pressemitteilung. Die Klasse ist also vielseitig einsetzbar. Zusätzlich zu dem Titel und dem Text zu dem Artikel wird das Text-Artikel-Objekt noch mit folgenden Eigenschaften und Komponenten versehen:

- Eine Liste aller Wörter (inklusive Stoppwörter) [202]
- Eine Inventarliste aller signifikanten Wörter (exklusive Stoppwörter)
- Eine Liste aller Sätze des Textartikels[203]
- Anzahl der Wörter

[199] Es wurden nur Tabellen übernommen, die Marktstudien speichern
[200] In diesem Fall Marktstudien
[201] s. S. 56 K. 4.6.1
[202] Wortliste
[203] Satzliste

- Anzahl der signifikanten Wörter
- Anzahl der Sätze
- Eine Liste aller Wörten und ihrer Frequenz bzw. absoluten Häufigkeit im Textartikel[204]

Folgendes UML Diagramm veranschaulicht die Schnittstelle und ihre Realisierung durch eine Klasse:

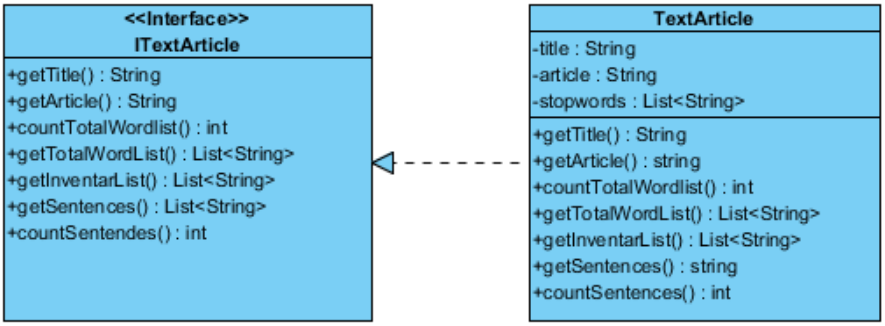


Abbildung 6.1: Schnittstelle und Klasse zu *ITextArticle* bzw. *TextArticle*

Für die Erstellung eines *Textabstract* reicht diese Klasse nicht aus, da sie nur die Textartikel darstellt. Um die Texte zu analysieren und zu einem *Abstract* zusammenzufassen, bedarf es noch einer Analyse Klasse. Grundlage dafür ist die Schnittstelle *IArticleAnalyzer*. Diese Analyse Klasse muss folgende Aufgaben erfüllen:

- Berechnung der Satzsignifikanz
- Operator zur Ausgabe des *Abstract*

Auf Basis dieser Schnittstelle werden die Extraktionsmethoden von Luhn und Edmundson realisiert:

[204] Wortfrequenz Liste

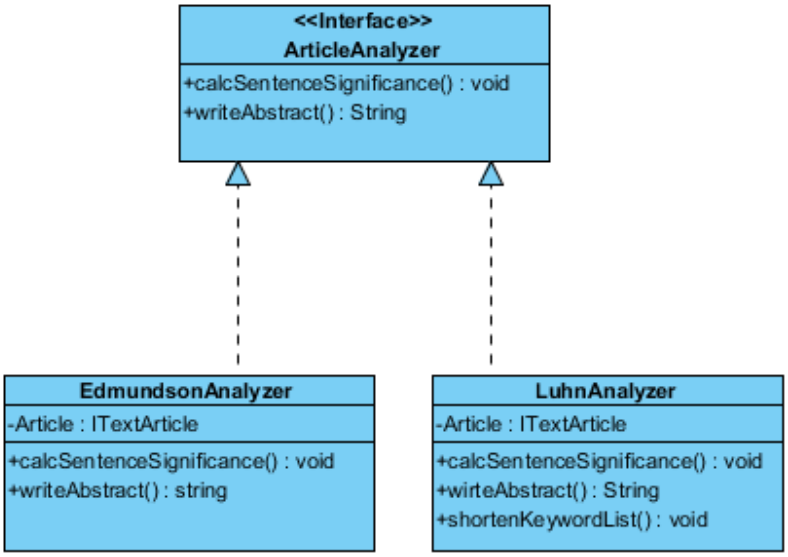


Abbildung 6.2: Die Beziehungen zwischen den einzelnen *Article Analyzern*

Sowohl der *LuhnAnalyzer* als auch der *EdmunsonAnalyzer* verwenden als Grundlage das vom *ITextArticle* instanziierte *TextArticle* Objekt. Wichtig hierbei ist die Übergabe des *TextArticle* Objekts an den Konstruktor der *Luhn-* bzw. *EdmundsonAnalyzer* Klasse. Der Unterschied zwischen den beiden *Analyzern* liegt in der Art und Weise, wie die Satzsignifikanz berechnet wird. Die ausführliche Beschreibung über die Bestimmung der Satzsignifikanz wird in den Kapiteln 6.3 bzw. 6.4 näher erläutert.

Eine letzte wichtige Komponente für die spätere Implementierung stellt die Schnittstelle *IWord* dar. In der *TextArticle* Klasse erfolgt die Berechnung der absoluten Häufigkeit einzelner Wörter im Text. Aus diesem Grund benötigt man einen Container, der *Wort-Objekte* beinhaltet, die ein Wort und ihre zugehörige absolute Häufigkeit repräsentieren. Die Schnittstelle *ITextArticle* benutzt *IWord* als Referenz für die berechneten Wortfrequenzen.

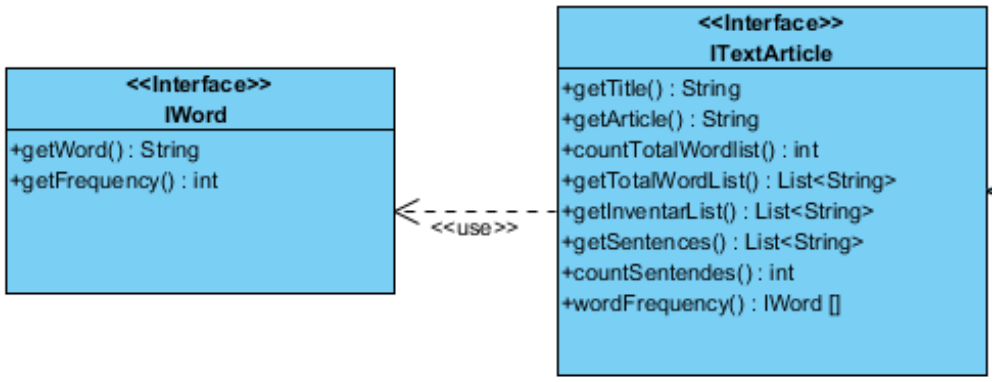


Abbildung 6.3: Die von *ITextArticle* verwendete Schnittstelle *IWord*

Der Operator *wordFrequency()* aus *ITextArticle* liefert einen Container, der die *Wort-Objekte* enthält, zurück. Im *IWord* gelangt man über die Operatoren *getword()* und *get-Frequency* auf das Wort bzw. auf die absolute Häufigkeit des Wortes. Die *IWord* Komponente ist für die weitere Entwicklung der Anwendung elementar und unverzichtbar, da auf Basis von *IWord* statistische Berechnungen durchgeführt werden. V. a. kommt es zur Berechnung der Häufigkeitsklasse bei der Erstellung des Kategoriendictionary zum Einsatz.

6.2. Die elementaren Komponenten eines Textes

Allgemein betrachtet besteht ein Text bzw. Textartikel aus Sätzen und Wörtern. Da Texte i. d. R. unstrukturiert sind, erhält der Text durch die Einteilung der vorhin genannten Komponenten eine gewisse Struktur. Hauptverantwortlich für die Zerlegung des Textes in Sätze und Wörter ist die Schnittstelle *ITextArticle*. Diese Informationen sind später für die Erstellung des *Abstract* von enormer Bedeutung, da auf Grundlage der Sätze bzw. Wörter die Satz- bzw. Wortsignifikanz berechnet wird. Aus Sicht der Implementierung werden diese Daten dem *LuhnAnalyzer* bzw. dem *EdmundsonAnalyzer* übergeben.

6.2.1. Die Segmentierung nach Wörtern

Die Zerlegung eines Textes in einzelne Wörter ist relativ trivial. Wörter werden i. d. R. durch ein Leerzeichen abgegrenzt und anhand dieses Leerzeichens erfolgt letztendlich die Zerteilung des Textes in Wörter.

```
1 Prozedur erstelleWortListe:
2 Input: Text
```

```
Output: die komplette Wortliste
Container wörter[]=trenneNachLeerzeichen(Text);
n:=Grösse des Containers wörter[];
Liste m;

for i=0 to n-1 do
  //speichere die einzelne Wörter in List m
  m:=wörter[i];
end for
return m;
```

Listing 22: Prozedur zur Zerlegung des Textes in Wörter

Als Ergebnis erhält man die komplette Wortliste eines Textes.

Bei der Erstellung einer Inventarliste geht man ähnlich vor, allerdings werden dabei die sogenannten Stopp- bzw. Funktionswörter nicht berücksichtigt. Typische Stoppwörter in der deutschen Sprache sind bestimmte Artikel[205], unbestimmte Artikel[206] und Konjunktionen[207]. Die Stoppwörter in der englischen Sprache sind äquivalent zu denen in der deutschen. Folgende Auflistung zeigt die zehn am meisten verwendeten Stoppwörter in der englischen Sprache[208]:

- of
- to
- and
- a
- in
- for
- is
- the
- was
- that
- on

[205] der, die, das
[206] einer, eine, ein, einen
[207] und, oder
[208] http://wortschatz.uni-leipzig.de/html/wliste.html

Stoppwörter treten in Texten sehr häufig auf, übernehmen aber lediglich eine grammatikalische und syntaktische Funktion. Laut dem *Projekt Deutscher Wortschatz* der Universität Leipzig[209] [210]ist der bestimmte Artikel „der“ das am häufigsten verwendete Stoppwort. Entsprechend dazu ist im englischen der Artikel „the“ das häufigste Wort[211]. Die Erzeugung einer Inventarliste verläuft ähnlich wie das Indexieren eines Dokuments. Aufgenommen werden nur diejenigen Wörter, die tatsächlich den Dokumenteninhalt repräsentieren[212]. Stoppwörter tragen nicht zur Relevanz des Dokumenteninhalts bei. Da bei der Implementierung zunächst die englischen Marktstudien berücksichtigt werden, verwendet man zunächst eine englische Stoppwortliste, die man unter `http://members.unine.ch/jacques.savoy/clef/index.html` finden und verwenden kann[213]. Diese Stoppwörter werden in einer Textdatei hinterlegt. Bei der Erzeugung der Inventarliste wird diese Datei verwendet, um die Stoppwörter herauszufiltern. Folgender Pseudocode veranschaulicht die Prozedur:

```
1  Prozedur erstelleInventarListe:
2  Input: Text
3  Input: Datei in der die Stoppwörter hinterlegt sind
4  Output: die Inventarliste

6  //Datei auslesen und in eine Liste stopwort packen
7  Liste stopwort:=readFile(stopwort-datei);
8  Container wörter[]=trenneNachLeerzeichen(Text);
9  n:=Grösse des Containers wörter[];

11 Liste inventar_list;

13 for i=0 to n-1 do
14   IF wörter[i] nicht in Liste stopwort THEN
15     inventar_list:=wörter[i];
16   END IF
17 end for
18 return inventar_list;
```

Listing 23: Prozedur zur Erzeugung einer Inventarliste

[209] `http://wortschatz.uni-leipzig.de/`
[210] `http://wortschatz.uni-leipzig.de/html/wliste.html`
[211] Zwar steht in der englischen Sprache das Wort „of“ auf Platz 1 der am meisten gebrauchten Wörter, jedoch wird in K. 6.3 nachgewiesen, dass der bestimmte Artikel „the“ in englischen Marktstudien am häufigsten verwendet wird
[212] [SM83], S. 65
[213] Auf der beigelegten CD finden Sie die komplette englische Stoppwortliste

In Zeile 7 wird zunächst die Stoppwortdatei gelesen und in einer Liste gespeichert. Die weitere Prozedur verläuft ähnlich wie bei der Erzeugung der Wortliste. Der Unterschied liegt darin, dass zunächst in Zeile 14 abgefragt wird, ob das Wort ein Stoppwort ist und falls dies nicht der Fall ist wird das Wort übernommen.

Sowohl bei der Wortliste als auch bei der Inventarliste muss darauf geachtet werden, dass während der Erstellung keine Sonderzeichen wie Punkt oder Fragezeichen in die Liste aufgenommen werden. Zumeist bringen Wörter, die am Ende eines Satzes stehen, solche Sonderzeichen mit sich. Vor der Speicherung in die Liste muss das Wort also von diesen Zeichen befreit werden[214].

6.2.2. Die Segmentierung nach Sätzen

Im Gegensatz zur Zerlegung in Wörter verfolgt die Segmentierung des Textes nach Sätzen einen anderen Ansatz. Ein Satzende wird i. d. R. durch einen Punkt gekennzeichnet. Man würde also, einen Text anhand der Punkte (Satzzeichen) in Sätze aufteilen. Das Problem hierbei ist, dass Abkürzungen in diesem Fall fälschlicherweise als Satzende interpretiert werden. Marktstudien v. a. im medizinischen und pharmazeutischen Bereich verwenden in den Beschreibungen häufig auch Abkürzungen. Zur Umgehung dieses Problems wird zusätzlich zu einer Stoppwortdatei auch eine Datei benötigt, die alle gängigen Abkürzungen beinhaltet. Ähnlich wie die Stoppwörter werden auch die Abkürzungen in der Implementierung in einer Liste gespeichert. Bei der Extrahierung der einzelnen Sätze muss der Text Wort für Wort durchgegangen werden. Wenn ein Punkt an ein Wort anschliesst und es sich hierbei nicht um eine Abkürzung handelt, so wird dies als ein Satz interpretiert, der in einer Satzliste gespeichert wird. Das Prozedere wiederholt sich beim nächsten vorkommenden Punkt. Ein Satz muss nicht immer mit einem Punkt enden, auch Fragezeichen und Ausrufezeichen signalisieren ein Satzende. Auf diese Satzzeichen muss folglich auch geachtet werden.

```
1 Prozedur erstelleEineSatzListe:
2 Input: Text
3 Input: Datei in der alle gängigen Abkürzungen hinterlegt sind
4 Output: Eine Liste von allen Sätzen

6 //Datei auslesen und in eine Liste Abkuerzungen packen
7 Liste Abkuerzungen:=readFile(Abkuerzung-datei);
8 Container wörter[]=trenneNachLeerzeichen(Text)
9 n:=Grösse des Containers wörter[];
```

[214] Auf der beigelegten CD finden Sie eine Auflistung aller Sonderzeichen

```
11 Liste satz_liste;

13 //Man gehe jetzt wort für wort durch
14 satz:=leerer String;

16 for i=0 to n-1 do

18   /*Zwischenspeicher für die Sätze im Text*/
19   satz = satz + wörter[i];

21   IF wörter[i] endetMit(.) OR endetMit(?) OR endetMit(!) THEN

23     IF wörter[i] nicht in Liste Abkuerzungen THEN
24       satz_liste:=satz;
25       satz:=leerer String;
26     END IF

28   END IF
29 end for
30 return satz_liste;
```

Listing 24: Prozedur zur Erzeugung der Sätze

Eine Liste aller englischen Abkürzungen kann man über `http://www.indiana.edu/~letrs/help-services/QuickGuides/oed-abbr.html` einsehen. Folgende Auflistung zeigt exemplarisch einige gängige Abkürzungen in englischen Texten:

- Prof.
- Dr.
- U.S.
- ltd.
- co.
- p.m.
- a.m.
- ca.
- Va.

6.3. Die Implementierung der Luhn Methode

Luhns Methode zur Erstellung eines *Abstract* basiert auf statistischen Verfahren. Zunächst gilt es, über den *ITextArticle* eine Inventarliste von Wörtern zu erstellen. Anschliessend muss entschieden werden, welche der Wörter aus der Inventarliste eine relativ gute Signifikanz aufweisen. Die Inventarliste muss also noch einmal gekürzt werden, um nur die Wörter aufzunehmen, die eine hohe Aussagekraft besitzen. Verantwortlich für die Kürzung der Inventarliste ist im *LuhnAnalyzer* der Operator *shortenKeywordList()*[215]. Luhn schlug in seiner Methode vor, Wörter mit hoher und niedrieger Häufigkeitsfrequenz auszuschließen und nur die Wörter zu berücksichtigen, die eine mittlere Häufigkeit aufweisen[216]. Durch die in Listing 23 vorgestellte Methode zur Erstellung einer Inventarliste wurden bereits die Stoppwörter herausgefiltert, dies bedeutet, dass die Wörter mit hoher Häufigkeit bereits ausgeschlossen wurden. Übrig bleiben letztendlich Wörter mit niedriger Häufigkeit, Wörter mit mittlerer Frequenz und Wörter, die zwar oft vorkommen, jedoch keine Stoppwörter sind.

Bevor der *LuhnAnalyzer* zum Einsatz kommt, soll erst einmal Luhns Aussage bewiesen werden, dass Wörter mittlerer Häufigkeit eine hohe Signifikanz besitzen bzw. die Relevanz des Textinhalts wiedergeben. Für diese Beweisführung wird *ITextArticle* herangezogen, da in dieser Systemkomponente die Wörter in einem Text berechnet werden[217]. Als Testdaten werden Marktstudien aus reports-research.com entnommen. Folgende Tabelle zeigt einen Ausschnitt einer Wortliste aus einer englischen Marktstudie über den Diabetes Markt in BRIC[218] [219] Ländern. Insgesamt besteht der Text aus 230 Wörtern und 5 Sätzen. Die Sortierung erfolgt absteigend nach der absoluten Häufigkeit. Zusätzlich dazu wird zu jedem Wort die Häufigkeitsklasse in der Tabelle angezeigt, um die Wörter zu klassifizieren und dem Nachweis von Luhns Aussage mehr Nachdruck zu verleihen. Die Häufigkeitsklasse (HKL) ist ein statistisches Maß, das die Gebrauchshäufigkeit eines Wortes im Text feststellt. Für eine erfolgreiche Berechnung der Häufigkeitsklasse wird das am häufigsten verwendete Wort als Vergleichsgrundlage genommen. In der deutschen Sprache wäre es das Wort „der“, im Englischen das Wort „the“. Die Formel für die Berechnung der Häufigkeitsklasse lautet folgendermaßen[220]:

> Es sei:
> h(w)=Anzahl des zu untersuchenden Wortes w

[215] s. S. 77 K. 6.1, Abb. 6.2
[216] s. S. 68 K. 5.4.1
[217] s. S. 77 K. 6.1, Abb. 6.1
[218] Brazil, Russia, India, China
[219] s. Anhang C, S. 131
[220] [HQW08], S. 97

h(the)=Anzahl von „the“

$$HKL(w) = 0.5 - \log_2(\frac{h(w)}{h(the)}) \tag{14}$$

Dabei gilt folgende Feststellung: Je kleiner der HKL-Wert, desto häufiger kommt das Wort vor. Wörter mit mittlerer bzw. geringerer Häufigkeit haben dann einen relativ höheren HKL-Wert. Wörter mit dem selben HKL-Wert kommen ungefähr gleich häufig vor.

Tabelle 1: Wortliste aus der Marktstudie „BRIC Diabetes Drugs Market“

Wort	Abs.Häufigkeit	HKL[221]
the	17	0
of	15	0
to	8	1
BRIC	7	1
in	6	2
market	6	2
diabetes	5	2
drugs	5	2
global	4	2
by	3	3
Markets	2	3
disease	2	3
CAGR	2	3
offering	2	3
Diabetes	2	3
Research	2	3
is	2	3
significant	2	3
account	2	3
prevalence	2	3
Market	2	3
condition	2	3
countries	2	3
Drugs	2	3
patients	1	4

Wort	Abs.Häufigkeit	HKL
Mercks	1	4
Galvus	1	4
Investor	1	4
Januvia	1	4
diabetic	1	4
patient	1	4
injectables	1	4
incretin	1	4
mimetics	1	4
inhibitors	1	4
Driven	1	4
Novartis	1	4
Eli	1	4
but	1	4
Oral	1	4
dipeptidyl	1	4
Lillys	1	4
drug	1	4
pharma	1	4
Byetta	1	4

Anhand dieser Tabelle wird ersichtlich, dass das am häufigsten gebrauchte Wort, in diesem Fall die Stoppwörter „the“ und „of“, einen HKL-Wert von 0 besitzt und deshalb ganz oben in der Rangliste steht. Der HKL-Wert der relevanten Begriffe steht zwischen 2 und 3, so z. B. bei den Begriffen „Diabetes“ und „drugs“, die das Kernthema des Textes widerspiegeln. Somit war Luhns Einschätzung, dass Wörter mit mittlerer Häufigkeit eine hohe Relevanz aufweisen vollkommen richtig.

Auf Grundlage dieser Wortliste wird im *ITextArticle* die Inventarliste erzeugt[222]. Im nächsten Schritt wird diese Liste mit dem *LuhnAnalyzer* noch einmal gekürzt. Bei diesem Vorgang wird für jedes Wort seine relative Häufigkeit[223] berechnet, um die Signifikanz des Wortes im Verhältnis zum gesamten Text zu bestimmen. Da die Stoppwörter schon herausgefiltert wurden, ist die Berechnung der relativen Häufigkeit ein geeigneter Ansatz, um aus den restlichen Wörtern die relevantesten herauszufiltern, jene also, die für die auto-

[222] s. S. 80 K. 6.2.1, List. 23
[223] Termfrequenz

matische Zusammenfassung geeignet sind. Die mathematische Formel[224] zur Ermittlung der relativen Häufigkeit ist aus der Statistik bekannt:

Es sei:
h(w)=Häufigkeit eines Wortes im Dokument
a(d)=Anzahl aller Wörter im Dokument

$$TF(w,d) = (\frac{h(w)}{a(d)}) \tag{15}$$

Als Beispielrechnung wird das Wort „diabetes“ aus der Tabelle 1 entnommen:

Anzahl von „diabetes“ = 5
Anzahl aller Wörter = 230

$$TF(w,d) = (\frac{5}{230}) \tag{16}$$

$$TF(w,d) = 0,00217 \tag{17}$$

Ggf. erhält man bei langen Texten sehr kleine Werte. Um den Ergebnisbereich ein wenig einzuengen, wird die Formel durch einen Logarithmus noch einmal verfeinert[225]:

$$TF(w,d) = (\frac{\log_2(5)}{\log_2(230)}) \tag{18}$$

$$TF(w,d) = 0,29 \tag{19}$$

Die so berechneten Werte dienen als Indikator dafür, ob ein Wort für die Zusammenfassung überhaupt geeignet ist. Es muss ein Schwellenwert ausgewählt werden, ab dem ein Wort als geeignet betrachtet werden kann. In [SCH08] wird der Schwellenwert von 0,2 empfohlen. Folgende Tabelle veranschaulicht in Auszügen die Wörter aus der Inventarliste mit ihrer relativen Häufigkeit. Die Wörter aus der Tabelle 1 dienen als Grundlage.

[224] s. [SCH08]
[225] ebenda

Tabelle 2: Wörter aus der Inventarliste mit relativer Häufigkeit

Wort	Rel. Häufigkeit
BRIC	0,35783041006083205
diabetes	0,29595705045956533
drugs	0,29595705045956533
global	0,2549235276589582
.....	.....
Markets	0,1274617638294791
announced	0,1274617638294791
.....	.....
rising	0,0
incidence	0,0
pharma	0,0

Mit dem Schwellenwert von 0,2 würden die ersten vier Wörter aus der Tabelle 2 in die gekürzte Liste übernommen. Bei genauerer Betrachtung sind es jene Wörter, die den Kontext des Textes wiedergeben und somit für eine Zusammenfassung geeignet sind. Der Schwellenwert wird durch die Länge des Textes determiniert. Die folgende Tabelle veranschaulicht diese Sachlage und zeigt eine Wortliste aus einem Text mit 101 Wörtern und 3 Sätzen:[226].

Tabelle 3: Wörter aus der Inventarliste mit relativer Häufigkeit aus einem Text mit 101 Wörtern

Wort	Rel. Häufigkeit
body	0,3003809664473759
drugs	0,15019048322368794
.....	.....
hormone	0,15019048322368794
substances	0,15019048322368794
.....	.....
recreational	0,0
beers	0,0
wines	0,0
insulin	0,0

Setzt man hier den Schwellenwert auf 0,2, dann würde nur ein Wort in die gekürzte Inventarliste übernommen werden. Dies könnte dazu führen, dass überhaupt keine Zusammenfassung zurückgeliefert wird. Der Schwellenwert muss in diesem Fall auf 0,1 reduziert werden. Für die spätere Anwendung bedeutet dies, dass der Schwellenwert bei der Erstellung eines *Abstract* variabel sein muss.

[226] s. Anhang C, S. 134

6.3.1. Hintergrund der Luhn Methode

Die im vorherigen Kapitel vorgestellten Methoden basieren auf dem Zipfschen Gesetz. Die Verwendung der natürlichen Sprache folgt i. d. R. dem „Prinzip der geringsten Anstrengung“[227], das besagt, dass Autoren oder Sprecher aus praktischen Gründen bestimmte Wörter wiederholen, anstatt permanent nach neuen zu suchen[228]. Aus diesem Grund sind die am häufigsten gebrauchten Wörter Stoppwörter. Dies wurde in der Tabelle 1 des vorherigen Kapitels nachgewiesen. Desweiteren stellt Zipf fest, dass die Häufigkeit, in der ein Wort auftritt, umgekehrt proportional ist zu seinem Rang. Durch das Zipfsche Gesetz, erhält man einen Ansatzpunkt, um *Wortbedeutsamkeitsfaktoren*[229] herzuleiten.

6.3.2. Die Berechnung der Satzsignifikanz

Nach der Kürzung der Inventarliste muss nun anhand dieser die Zusammenfassung durchgeführt werden. Dabei werden die Sätze des Textes benötigt, die sich ebenfalls in einer Liste befinden[230]. Für die Berechnung der Satzsignifikanz wird Luhns Formel angewendet[231]. Zur Erinnerung zeigt folgendes Listing Luhns Formel noch einmal auf.

Es sei:
SignificanceWords = Anzahl der signifikanten Wörter innerhalb eines Satzes
AllWords = Anzahl der Wörter innerhalb eines Satzes
Sigfactor = Signifikanzfaktor des Satzes

$$Sigfactor = \frac{(SignificanceWords)^2}{AllWords} \tag{20}$$

Die Formel beruht auf der Tatsache, dass zwischen zwei signifikanten Wörtern nicht mehr als 3 oder 4 nicht-signifikante Wörter stehen. Da diese Formel pro Satz angewendet werden muss, wird die Satzliste durchlaufen und jedes Wort im Satz mit dem Wort in der gekürzten Liste verglichen[232]. Um dies korrekt durchzuführen werden pro Satz die

[227] [HQW08], S. 87
[228] [SM83], S. 66
[229] ebenda, S. 66
[230] s. S. 83 K. 6.2.2, List. 24
[231] s. S. 68 K. 5.4.1
[232] vgl. ebenda

Wörter extrahiert, ähnlich wie bei der Erstellung einer Wortliste[233]. Der folgende Pseudocde verdeutlicht die Vorgehensweise:

```
Prozedur calcSentenceSignificance:
Input: Liste aller Sätze
Input: gekürzte Inventarliste
Output: Der Wert der Satzsignifikanz

Liste Saetze:=satz_liste;
Liste inventarliste:=inventar_liste;

for each satz in saetze

  Conteiner wörter[]:=trenneNachLeerzeichen(satz);
  n:=Grösse des Containers wörter[];

  SignifikantesWort:=0;

    for i=0 to n-1 do

      if wörter[i] in Liste inventarlist THEN
        //hochzählen
        SignifikantesWort++;
      end if


    end for

    //Wende die Luhn Formel an.
    Ergebniss:=berechneSignifikanzNachLuhn(SignifikantesWort^2 / n);
    Ausgabe Ergebniss;

end for each
```

Listing 25: Prozedur zur Berechnung der Satzsignifikanz

Als nächstes muss ein geeigneter Schwellenwert bestimmt werden, der ausschlaggegebend ist für die Aufnahme in eine Zusammenfassung. Er darf nicht negativ sein oder gegen Null tendieren. Um einen guten Wert herauszufinden, müssen empirische Beobachtungen durchgeführt werden. Als Testdaten verwendet man erneut die Beispielmarktstudie aus dem Kapitel 6.3. Folgende Tabelle zeigt in Auszügen die Satzsignifikanzen aus der Beispielmarktstudie.

[233] vgl. S. 80 K. 6.2, List. 22

Tabelle 4: Satzsignifkanz Marktstudie „BRIC Diabetes Drugs Market“

Satz-Nr.	Satz	Signifikanz
1	Research and Markets has announced the addition of the BRIC Diabetes Drugs Market report to their offering[. . .].	0,058
2	BRIC countries hold a significant share of the global market for diabetes drugs[. . .].	1,113
3	Living standards the rising incidence of diabetes in BRIC[. . .].	1,35
4	The slowdown of the pharma market growth in developed economies has concentrated[. . .].	0,438
5	The BRIC diabetes drugs market is expected to grow from[. . .].	1,63

Anhand dieser Tabelle erkennt man, dass nur solche Sätze interessant sind, die eine Satzsignifikanz von größer oder gleich 1 besitzen. Bei diesen Sätzen tauchen die wichtigen Wörter wie „BRIC“ oder „diabetes“ auf, die den Kontext wiedergeben. Der Signifikanzwert hängt von der Länge des Textes und von der Qualität der Inventarliste ab. Jedoch ist der Wert 1 ein guter Einstiegspunkt, um einen Grenzwert zu bestimmen. Der Pseudocode in Listing 25 muss daher noch einmal angepasst werden.

```
1  Prozedur calcSentenceSignificance: Input: Liste aller Sätze
2  Input: gekürzte Inventarliste
3  Output: Liste der Signifikantesten Sätze

5  Liste Saetze:=satz_liste;
6  Liste inventarliste:=inventar_liste;
7  Liste SignifikanteSätze;

9  for each satz in saetze

11   Conteiner wörter[]=trenneNachLeerzeichen(satz);
12   n:=Grösse des Containers wörter[];

14   SignifikantesWort:=0;

16     for i=0 to n-1 do

18       if wörter[i] in Liste inventarlist THEN
19         //hochzählen
20         SignifikantesWort++;
```

```
21      end if


24    end for

26    //Wende die Luhn Formel an.
27    Ergebniss:=berechneSignifikanzNachLuhn(SignifikantesWort^2 / n);

29    IF Ergebniss >= 1 then
30      //Auswahl von signifikanten Sätzen
31      SignifikanteSätze=satz;
32    END IF


35 end for each

37 return SignifikanteSätze
```

Listing 26: Erstellung einer Satzsignifikanzliste

Es wird noch einmal überprüft, ob die Satzsignifikanz einen Mindestwert von 1 besitzt und falls dies zutrifft, wird der Satz in die Liste übernommen. Die komplette Signifikanzliste wird dann letztendlich dem Operator *writeAbstract* aus dem *LuhnAnalyzer* übergeben. Dieser gibt anschliessend ein *Abstract* aus[234].

6.4. Die Implementierung der Edmundson Methode

Die Methode von Edmundson verfolgt einen anderen Ansatz als die Methode von Luhn[235]. Ein wichtiger Bestandteil der Edmundson Methode ist das *Cue Dictionary*[236], in dem fachspezifische Begriffe enthalten sind. Für die Berechnung der Satzsignifikanz nach Edmundson ist dieses Dictionary elementar. Zur Realisierung des *Cue Dictionary* wird eine MySQL-Datenbank verwendet. In dieser Datenbank stellen zwei Tabellen das *Cue Dictionary* dar: Eine Tabelle, in der die Quelle bzw. der Text, aus dem die *Cue Wörter* stammen, hinterlegt wird und eine weitere Tabelle, das *Cue Dictionary* selbst.

[234] s. S. 77 K. 6.1, Abb. 6.2
[235] s. S. 74 K. 5.4.3
[236] s. S. 71 K. 5.4.2

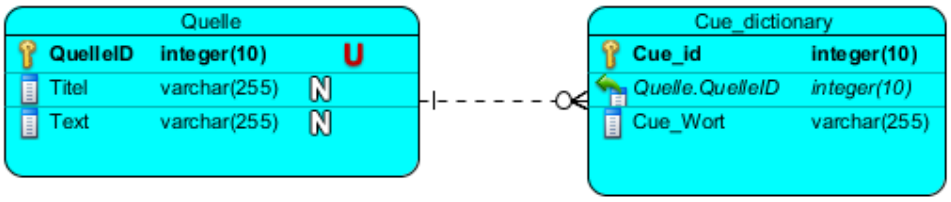


Abbildung 6.4: Datenbankstruktur zum *Cue Dictionary*

Prinzipiell muss am Anfang die *Cue Dictionary* Tabelle mit Wörtern gefüllt werden. Edmundson versorgte damals das System mit fachspezifischen Dokumenten, was im Falle des Marktstudienportals bedeutet, dass das *Cue Dictionary* mit Begriffen aus der Marktforschung versehen wird. Als „Lernmaterialien“ für das Dictionary eignen sich ein weiteres Mal die Marktstudiendaten aus reports-research.com[237]. Bei der Implementierung kommen wieder die Schnittstellen *ITextArticle* und *IWord* zum Einsatz. Für die endgültige Analyse und Erstellung des *Abstract* ist der *EdmundsonAnalyzer* verantwortlich. Da Martkstudien einer oder mehreren Kategorien zugeordnet sind, geht man so vor, dass man je Kategorie eine Marktstudie entnimmt und die Wörter aus der Beschreibung in dem *Cue Dictionary* speichert. Dabei muss beachtet, dass ggf. schon bekannte bzw. vorhandene Wörter im *Cue Dictionary* nicht überschrieben werden.

6.4.1. Die Berechnung der Satzsignifikanz

Erst nachdem das *Cue Dictionary* mit Daten versorgt wurde, kann mit dem Vorgang der Zusammenfassung begonnen werden. Bei der Berechnung der Satzsignifikanz klassifiziert Edmundson drei Wortklassen[238]:

- Bonuswörter
- Stigmawörter
- Nullwörter

Für die Zusammenfassung werden erneut die Sätze benötigt, die sich in der Satzliste befinden[239]. Neben dieser Satzliste wird ebenfalls die Stoppwortliste benutzt, um die Nullwörter zu identifizieren. Ähnlich wie bei Luhns Methode wird die Satzliste durchgegangen und jedes Wort mit denen im *Cue Dictionary* verglichen. Hierbei müssen demnach die Wörter aus den Sätzen extrahiert werden. Anders als bei Luhns Methode wird keine statistische Berechnung durchgeführt, es werden die Bonus-, Stigma- und Nullwörter ermittelt und gewichtet. Die Gewichtung erfolgt folgendermaßen:

[237] In der beigelegten CD finden Sie eine Linksammlung zu den Lernadaten aus www.reports-research.com
[238] vgl. S. 71 K. 5.4.2
[239] vgl. S. 90 K. 6.3.2

- Gewicht 1 bei Bonuswörtern
- Gewicht -1 bei Stigmawörtern
- Gewicht 0 bei Nullwörtern

Die so ermittelten Gewichtungen im Satz werden addiert, das Ergebniss stellt die Satzsignifikanz dar.

```
INPUT: stoppwortliste
INPUT: Cue wortliste aus der Cue Dictionary
INPUT: Text Article
OUTPUT: Signifikante Sätze

Liste stopwort_liste:=readFile(stopwort-datei);
Liste Cue_Wortliste:=SelectFromDataBase(Cue_Dictionary);
Liste Saetze:=satz_liste;

for each satz in Satze

 Conteiner wörter[]=trenneNachLeerzeichen(satz);
 n:=Grösse des Containers wörter[];

 gewicht:=0;
 Signifikanz:=0;
 for i=0 to n-1 do

   if wörter[i] in Cue_wortliste THEN

     if wörter[i] in stopwort_liste THEN
       gewicht:=gewicht + 0;
     else
       gewicht:=gewicht + 1;
     end if

   else
     gewicht:=gewicht - 1;
   end if

   Signifikanz:=Signifikanz + gewicht;

 end for

 Ausgabe satz und ihre Signifikanz;
```

```
38 end for each
```

Listing 27: Berechnung der Satzsignifikanz nach Edmundson

Äquivalent zu der Luhn Methode muss wieder ein geeigneter Schwellenwert ermittelt werden, ab dem ein Satz eine hohe Signifikanz aufweist. Die Testdaten werden diesesmal der Marktstudie „Diabetes Market in UAE“[240] entnommen. Folgende Tabelle zeigt die Satzsignifikanzen nach Edmundson:

Tabelle 5: Satzsignifkanz Marktstudie „Diabetes Market in UAE“

Satz-Nr	Satz	Signifikanz
1	Diabetes is one of the fastest growing lifestyle and debilitating diseases in the Middle East region[. . .].	-5
2	At present one out of every five person in the UAE is suffering from diabetes[. . .].	7
3	The concern becomes a bit serious as diabetes is associated with several other chronic diseases like cardiovascular diseases[. . .].	6
4	According to our new research report Diabetes Market in UAE the UAE diabetes care market[. . .].	63
5	This has put an extra burden on the countrys healthcare spending to allocate more funds for diagnosis care and prevention[. . .].	-66

Als Ergebnis erhält man entweder einen negativen oder einen positiven Signifikanzwert. Bei genauerer Betrachtung der Tabelle beinhalten die Sätze mit positivem Signifikanzwert die meisten relevanten Begtriffe wie „UAE“, „diabetes“ und „market“. Dies lässt die Schlussfolgerung zu, dass nur die Sätze mit positivem Signifikanzwert für eine Zusammenfassung geeignet sind.

6.5. Fazit

Beide o. g. Verfahren zielen bei einer Erstellung eines *Abstract* darauf ab, nur die signifikantesten Sätze auszuwählen. Dabei werden die in dem Text befindlichen Wörter zur

[240] s. Anhang C, S. 134

Hilfe genommen. Doch hier enden auch die Gemeinsamkeiten zwischen der Luhn und der Edmundson Methode, die Vorgehensweise bei der Auswahl der Satzsignifikanz ist bei beiden unterschiedlich. Es stellt sich nun die Frage, welche der beiden Methoden die geeignetere ist, Luhns statistisches Verfahren oder die Signalwortmethode nach Edmundson. Der Vorteil bei Luhn liegt in der Tatsache begründet, dass die Berechnung auf vorhandenen Modellen und Gesetzen wie dem Zipfschen Gesetz basiert. Auf Grundlage der schon vorhandenen Grundsätze kann man eine praxisbasierte Lösung herausarbeiten. Mit den dazugehörigen mathematisch-statistischen Verfahren erzielt man relativ gute Ergebnisse. Die Ergebnisse sind jedoch nicht immer exakt, gelegentlich muss mit Näherungswerten gearbeitet werden, so z. B. bei der Bestimmung eines Schwellenwertes. Es bedarf weiterer empirischer Beobachtungen, um den geeigneten Schwellenwert zu ermitteln[241]. Hier erhält man zumindest einen Ansatzwert, mit dem weiter gearbeitet werden kann. Im Gegensatz dazu besitzt die Edmundson Methode einen völlig anderen Ansatz, sie ist stark fachspezifisch und frei von statistischen Verfahren. Aus diesem Grund sind fachspezifische Begriffe von Nöten, um eine erfolgreiche Abstrahierung zu erzielen. Das Problem hierbei ist, dass Marktstudien grundsätzlich thematisch sehr breit gefächert sind. Die Fachspezifität ist in diesem Fall kategorienabhängig, was dazu führen könnte, dass das System zu viele Wörter „kennt“ und keine vernünftige Zusammenfassung erstellen kann, da ggf. die Signifikanzwerte aller Sätze positiv sind. Hierbei fehlt bei Edmundson eine Lösung, wie man aus vielen Fachgebieten die geeigneten Fachbegriffe spezifizieren kann. Dies rührt v. a. daher, dass bei Edmundson eben diese statistischen Methoden fehlen. Dieses Problem könnte jedoch mit einigen Methoden aus dem *Text Mining* behoben[242] werden.

[241] s. Anhang C, S. 131
[242] s. S. 102 K. 7.2.2

7. Die Kategorienwortliste

Die Kategorienwortliste ist eine wichtige Komponente, die bei der Kategorisierung eine zentrale Rolle einnimmt. Dieses Kapitel beschäftigt sich mit der Erstellung der Wortliste, es präsentiert eine geeignete Methode aus dem *Text Mining* und wendet diese an.

7.1. Die Tabellenstruktur

Die Wortliste wird wie bei Edmundsons *Cue Dictionary* in einer MySQL Datenbank gespeichert. Hierfür wird neben der Wortlistentabelle auch eine Kategorientabelle benötigt.

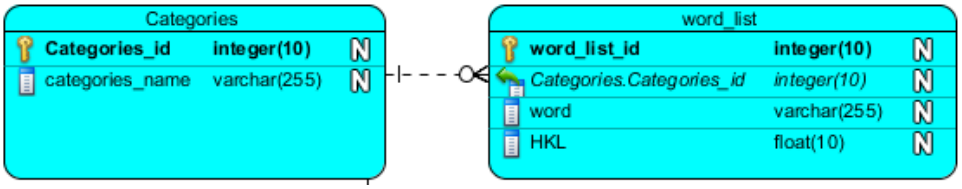


Abbildung 7.1: Die Tabellenstruktur für die Wortliste

7.2. Die Erstellung der Wortliste

Wichtigster Bestandteil für eine Erzeugung der kategorienbasierten Wortliste ist der sogenannte Korpus bzw. Dokumentenkorpus, der in diesem Fall durch die vorhandenen Marktstudien[243] der Testdatenbank repräsentiert wird. Der Dokumentenkorpus mit den dazugehörigen Marktstudien wird je Kategorie in XML hinterlegt und dadurch ist der Zugriff auf den Dokumentenkorpus plattformunabhängig[244]. Zusätzlich dazu wird auch ein Gesamtkorpus erzeugt, also ein Dokumentenkorpus mit allen Marktstudien, der auch in XML hinterlegt wird. Insgesamt existieren somit mehrere Kategorienkorpi und ein Gesamtkorpus.

Bei der Erzeugung der Wortliste kommen wieder die Schnittstellen *ITextArticle* und *IWord* zum Einsatz. Dabei werden die Stoppwörter, ähnlich wie bei der Erstellung einer Inventarliste, nicht berücksichtigt[245]. Auf Basis von *IWord*, das das Wort und seine absolute Häufigkeit beinhaltet und von *ITextArticle* verwendet wird, bestimmt man die Häufigkeitsklasse(HKL) eines Wortes im Dokumentenkorpus. Dabei werden die Marktstudien nach

[243] s. S. 56 K. 4.6.1
[244] s. S. 27 K. 3.1
[245] s. S. 77 K. 6.1, List. 23

Kategorien gewählt und in verschiedenen Kategorienkorpi gesammelt. Anhand dieser Sammlung werden die Wörter extrahiert und pro Wort wird die HKL ausgerechnet. Als Grundlage wurden vorerst knapp 400 englischsprachige Marktstudien in die Testdatenbank importiert. Für den ersten Testdurchlauf wurden die Kategorien „Healthcare“, „Food“ und „Beverages“ ausgesucht. Die Länge der Marktstudienbeschreibung variiert in den jeweiligen Kategorien.

Tabelle 6: Marktstudienanzahl und -wortliste nach Kategorien

Kategorie	Anzahl Marktstudien	Größe der Wortliste	kleinster HKL	größter HKL
Healthcare	39	2.859	2	10
Food	22	1.376	2	9
Beverages	8	973	2	8

In folgenden Tabellen werden in Auszügen einige Wörter der jeweiligen Wortlisten einander gegenübergestellt.

Tabelle 7: Ausgewählte Wörter aus „Healthcare“

Wort	HKL
market	2
treatment	5
diabetes	6
cardiovascular	7
pharmacologic	10
reviews	10

Tabelle 8: Ausgewählte Wörter aus „Food“

Wort	HKL
market	2
snacks	4
seed	5
coffee	6
diet	9
controlled	9

Tabelle 9: Ausgewählte Wörter aus „Beverages“

Wort	HKL
market	2
drinks	4
alcoholic	5
juice	6
nectar	8
reduction	8

Man sieht, dass in den Kategorien die interessantesten Wörter meistens eine mittlere HKL besitzen, aber auch Wörter mit höherer HKL sind in Ausnahmefällen relevant. Bei näherer Betrachtung der Tabellen taucht in jeder Wortliste der Begriff „market“ auf, der einen niedrigen HKL Wert besitzt, dies bedeutet, dass dieser Begriff in jeder Kategorie sehr häufig vorkommt. Legt man die Kategorienkorpi zu einem Gesamtkorpus zusammen, so vervielfacht sich die Häufigkeit von „market“. Bei der Kategorisierung der Marktstudien anhand des *Abstract* kann dies zu einem Problem führen – die zu kategoriesierende Marktstudie wird zu vielen Kategorien zugeordnet, weil sie das Wort „market“ enthält. Ziel ist die Verfeinerung der Kategorienwortliste. Ein Wort erhält erst dann eine besondere Gewichtung, wenn es in einigen Dokumenten besonders häufig, im Allgemeinen jedoch eher selten auftaucht[246]. Die inverse Dokumentfrequenz liefert gemeinsam mit der Termfrequenz diese Einteilung[247].

7.2.1. Die TF-IDF

Die Termfrequenz und inverse Dokumentfrequenz, oder kurz TF-IDF, liefert für jedes Wort ein statistisches Maß für seine Relevanz. Grundsätzlich ist die Bedeutsamkeit eines Wortes „[. . .] proportional zur Häufigkeit des Begriffes k im Dokument und umgekehrt proportional zur Gesamtzahl der Dokumente, in denen der Begriff [. . .]“[248] gefunden wurde. Die Berechnung der Termfrequenz ist ähnlich der Berechnung der relativen Häufigkeit, mit dem Unterschied, dass in diesem Fall statt eines einzelnen Dokuments mehrere Dokumente berücksichtigt werden müssen. Die Termfrequenz ist der Quotient aus der Anzahl eines bestimmten Wortes im Gesamtdokument und der Summe aller Wörter aus den Dokumenten, in dem sich das bestimmte Wort befindet. Die inverse Dokumenthäufigkeit hingegen wird aus dem Quotienten der Dokumentenanzahl und der Anzahl der Dokumente, in denen das Wort gefunden wurde, gebildet. Folgende Formel veranschaulicht

[246] [HQW08], S. 204
[247] TF-IDF
[248] [SM83], S. 68

die TF-IDF[249]:

Es sei:
d=das Dokument
w=das bestimmte Wort
N=Summe aller Wörter aus den Dokumenten, in denen das bestimmte Wort gefunden wurde

$$tf = \frac{|w|}{N} \tag{21}$$

$$idf = \log \frac{|d|}{|d : w \in d|} \tag{22}$$

$$tf - idf = tf \cdot idf \tag{23}$$

Das Ergebnis der Rechnung indiziert die Gewichtung eines Wortes im Korpus. Um geeignete Wörter in die Wortliste aufzunehmen, muss ein Schwellenwert definiert werden, ab dem ein Wort eine Relevanz in Bezug auf die Kategorie besitzt. Je höher der TF-IDF Wert ist, desto relavanter ist das Wort.

Tabelle 10: Auszug einer Wortliste aus „Healthcare“mit TF-IDF

Wort	TF-IDF
epigenetic	0,03948051134060792
stem	0,021563078532949066
diabetes	0,011266797318846633
market	0,0017529746508744123
treatment	0,001668256135383435
pharmaceutical	0,0011669975990736131
key	4,360543819544218E-4
analysis	4,452384329254626E-4

Nach dieser Tabelle haben relevante Wörter einen mindest TF-IDF Wert von 0,01. Man könnte also diesen Schwellenwert als Mindestgrenze einsetzen. Allerdings werden dadurch andere Wörter, die für die Kategorie eigentlich sinnvoller sind, aber einen niedrigeren TF-IDF Wert besitzen, ignoriert. Daher soll zusätzlich noch ermittelt werden wie sich ein Wort zum Gesamtkorpus verhält, also die Berechnung des TF-IDF Wertes auf Basis der kompletten Marktstudiendaten.

[249][HQW08], S. 205

Tabelle 11: Wörter aus „Healthcare“und deren TF-IDF Werte in Bezug zum Gesamtkorpus

Wort	TF-IDF
epigenetic	0,06437698636167312
treatment	0,0032779401611249567
pharmaceutical	0,0052548919555207695

Man erkennt, dass beispielsweise der TF-IDF Wert von „epigenetic“ im Gegensatz zu Tabelle 10 angestiegen ist, das gilt auch für „pharmaceutical“. Dies rührt v. a. daher, dass die Anzahl der Dokumente im Gesamtkorpus viel größer ist als im kategorienabhängigen Korpus. Folglich würden mit dem Zuwachs von Marktstudiendaten oder auch Dokumenten die TF-IDF Werte eines Wortes ansteigen. Es ist daher sinnvoller, die TF-IDF Berechnungen in der automatischen Kategorisierung einzusetzen, da hier der komplette Korpus gebraucht wird und somit Wörter, die im kategorienabhängigen Korpus zwar eine niedrige TF-IDF aufweisen, in Bezug auf den Gesamtkorpus aber einen höheren Seltenheitsgrad[250] und folglich einen höheren TF-IDF Wert besitzen. Auf die Verfeinerung der Kategorienwortliste wird vorerst verzichtet, da die Wortliste als Referenz Dictionary ausreicht. Das Hauptaugenmerk soll auf der automatischen Kategorisierung auf Basis der Ergebniss der *Text Extraction* liegen[251].

In der Implementierung ist die Schnittstelle *ICalcTF_IDF* verantwortlich für die Berechnung der TF-IDF Werte:

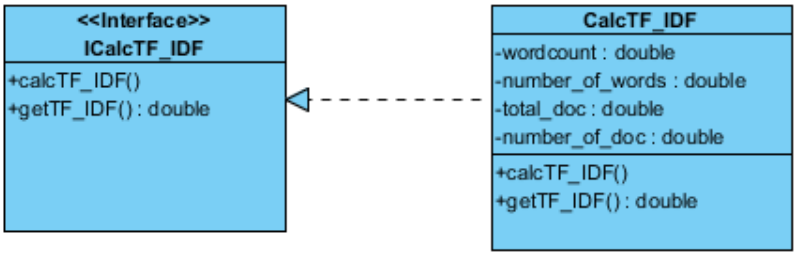


Abbildung 7.2: Die Schnittstelle *ICalcTF_IDF* ist verantwortlich für TF-IDF Berechnungen

7.2.2. Die Verbesserung von Edmundsons *Cue Dictionary*

Mit Hilfe der TF-IDF kann die Erstellung des Edmundson *Cue Dictionary* erheblich verbessert werden. Durch eine strengere Wahl eines TF-IDF Schwellenwertes wird das *Cue Dictionary* nur durch fachspezifische Begriffe gefüllt. Im Gegensatz zur Auswahl einer

[250] Vorausgesetzt es werden Marktstudien aus anderen Branchen als Healthcare importiert
[251] s. S. 104 K. 8

Marktstudie pro Kategorie als „Lernmaterialien“[252], verwendet man nun mehrere Marktstudien pro Kategorie. Durch die strenge Auswahl eines Schwellenwertes erhält man ein erheblich besseres fachspezifisches *Cue Dictionary*. Somit ergänzt man Edmundsons Signalwortverfahren um eine statistische Methode.

[252] vgl. S. 93 K. 6.4

8. Die automatische Kategorisierung anhand des *Abstract*

Nach der *Text Extraction* zur Erzeugung eines *Abstract* stellt die automatische Kategorisierung die Kernaufgabe der Anwendung dar und ist der letzte Schritt vor der Übertragung der Daten in OWL. An dieser Stelle sei noch darauf hingewiesen, dass der Vorgang bei genauerer Betrachtung semi-automatisch verläuft, d. h. die Anwendung schlägt auf Basis des *Abstract* Kategorien vor, die für eine Marktstudie geeignet sind. Anhand dieses Vorschlags kann dann der Benutzer selbst entscheiden, welche Kategorie angemessen ist. Ein weiteres Grundelement für einen fehlerlosen Ablauf der Kategorisierung ist die Kategorienwortliste. Zwischen dem *Abstract* und der Suche eines Wortes in der Kategorienwortliste kommt die Berechnung der TF-IDF zum Einsatz. Das Ziel ist es, die TF-IDF Werte der übrig gebliebenen Wörter im *Abstract* zu ermitteln. Hierbei nimmt der Gesamtkorpus, in dem sich alle Dokumente von Marktstudien befinden, eine wichtige Rolle ein, da auf seiner Grundlage die Berechnung der TF-IDF Werte durchgeführt wird. Man muss vorher einen Schwellenwert bestimmen, ab dem ein Wort in Relation zum Gesamtkorpus eine besondere „Singularität“[253] aufweist. Im Großen und Ganzen verläuft die Anwendung in folgenden Schritten:[254]

1. Lese eine Marktstudie ein[255].
2. Erstelle unter Zuhilfenahme des *LuhnAnalyzer* oder *EdmundsonAnalyzer* einen *Abstract*.
3. Die im *Abstract* befindlichen Wörter auflisten.
4. Bestimmung des TF-IDF Werts je Wort in Relation zum Gesamtkorpus.
5. Wörter übernehmen, deren TF-IDF Werte gleich oder über dem vorher bestimmten Schwellenwert liegen.
6. Diese Wörter in der Kategorienwortliste nachschlagen.
7. Die gefundenen Kategorien als Vorschlag anzeigen.
8. Überführung der zu kategorisierenden Marktstudien in OWL[256].

Technisch gesehen bedarf es für die Schritte 4 und 5 noch einer Liste aller Wörter aus dem *Abstract*, einer Liste aller Dokumente aus den Gesamtkorpus sowie der Kategorien-

[253] vgl. S. 67 K. 5.3
[254] Das Beispiel zeigt die Durchführung einer Kategorisierung von einer Marktstudie. Bei n Marktstudien wird dieser Vorgang n mal wiederholt
[255] Die Quelle kann eine XML Datei oder CSV Datei sein
[256] s. S. 109 K. 9

wortliste. Grundsätzlich duchläuft ein Wort den Gesamtkorpus, während dabei folgende Berechnungen ausgeführt werden[257]:

1. Die Anzahl des zu bestimmenden Wortes ermitteln.
2. Die Anzahl aller Wörter der Dokumente ermitteln, in der das bestimmte Wort sich befindet.
3. Die Anzahl aller Dokumente ermitteln.
4. Die Anzahl der Dokumente, in der das Wort gefunden wurde ermitteln.

Schritt 1 und 2 berechnen die Termfrequenz(TF), während 3 und 4 die inverse Dokumentfrequenz (IDF) bestimmen und schliesslich bilden alle vier Schritte zusammen die TF-IDF.

```
Input: Text aus Abstracts
Input: Gesamtkorpus
Output: vom System vorgeschlagenen Kategorienliste

Container wörter[]=trenneNachLeerzeichen(Abstract-Text)
n:=Grösse des Containers wörter[];
Liste dokumente:= readGesamtKorpus(Quelle);

/*Grösse des Korpus bzw. Anzahl der Dokumenten im Korpus*/
a:=AnzahlDokumente(dokumente);

s=:schwellenwert;



for i=0 to n-1 do

  anzahl_dok:=0;
  anzahl_wort:=0;
  anzahl_gesamt_wörter:=0;

  for each dokument in dokumente

    Container wörter_aus_dokument[]:=trenneNachleerzeichen(dokument);

    /*Anzahl der Wörter im aktuellen Dokument*/
    m:=Grösse des Containers wörter_aus_dokument[];

    for k=0 to k-1 do
```

[257] vgl. S. 100 K. 7.2.1

```
30      if wörter[i]==wörter_aus_dokument[k] THEN

32          /*Wenn wörter[i] in dokument gefunden dann inkrementieren*/
33          anzahl_wort:=anzahl_wort + 1;

35          /*Anzahl der gefundenen Dokumente, in denen wörter[i] auftaucht*/
36          anzahl_dok:=anzahl_dok + 1;

38          /*Anzahl aller Wörter aus dokumenten, in denen wörter[i] gefunden wurde*/
39          anzahl_gesamt_wörter:=anzahl_gesamt_wörter + m;

41      end if

43    end for


46  end for

48  //Berechne tf-idf
49  tf:=(anzahl_wort / anzahl_gesamt_wörter);
50  idf:=(a / anzahl_dok);
51  tf-idf:= tf * log(idf);

53  if tf-idf >=s then

55    /*in eine Liste aufnehmen*/
56    Liste rest_wörter:=wörter[i];

58  end

60 end for

62 Liste kategorie;

64 for each Wort in rest_wörter

66  kategorie = readKategorieListe(Wort);

68 end for

70 return kategorie;
```

Listing 28: Prozedur zur Berechnung der TF-IDF Werte

Beim Einlesen der einzelnen Dokumente aus dem Gesamtkorpus verwendet man wie bei den anderen Textanalysen die Schnittstellen *ITextArticle* und *IWord*.

8.1. Die Bestimmung des Schwellenwertes

Bei der Bestimmung des Schwellenwertes ist es sinnvoll, im Rahmen von Testfällen einige prägnante und gängige bzw. oft auftauchende Wörter auszuwählen und deren TF-IDF Wert zu bestimmen. Dabei schaut man, welche TF-IDF Werte die prägnanten und gängigen Wörter besitzen. Bei den Daten handelt es sich primär um Marktstudien, in denen spezielle Wörter wie „market“ oder „report“ häufig vorkommen. Im Grunde handelt es sich bei diesen Wörtern in Bezug auf die Marktstudiendaten ebenfalls um Stoppwörter. Um diese speziellen Stoppwörter auszuschliessen, muss überprüft werden, wie klein deren TF-IDF Werte sind und im Gegensatz dazu, wie hoch ein TF-IDF Wert eines prägnanten Wortes sein kann. Anhand dieser Beobachtungen wird ein geeigneter Schwellenwert abgeleitet. Wie in Kapitel 7.2 verwendet man die knapp 400 Dokumente als Testgrundlage. Später wird die Anzahl der Dokumente durch weitere Marktstudien erhöht, so dass beobachtet werden kann, ob und wie sehr sich der Schwellenwert ändert. In den ersten Testläufen wurden folgende Ergebnisse erzielt:

Tabelle 12: Auszug aus dem ersten Testdurchlauf mit 400 Dokumenten

Wort	TF-IDF
epigenetic	0,06437698636167312
milk	0,03202070199995807
bioinformatics	0,021415664870860104
livestock	0,020137091127160806
insulin	0,014710569796119455
metal	0,014240733956509757
beer	0,011721653089045601
diabetes	0,011460997630065688
food	0,009582538284157309
tools	0,008031373204019317
body	0,005515107429889175
pharmaceutical	0,0052548919555207695
introduced	0,003708735217884334
introduction	0,0036207770268922566
public	0,003415592477652624

Wort	TF-IDF
line	0,002916571062106434
forecast	0,0029079965737533035
report	0,002437779400987902
market	0,00199159224426669
inside	0,0016078599347192515
year	0,0013994780608796635
sector	0,0013527208793540212
revenue	7,273746357565604E-4
key	2,8869849374389933E-4

Im Großen und Ganzen entstammen die 400 Marktstudien aus der Testdatenbank den Bereichen *Heavy Industry*[258], *Food*[259], *Consumer Goods*[260], *Healthcare*[261], *Pharmaceutical*[262], *Life Science*[263]. Anhand dieser Tabelle erkennt man, dass die wirklich relevanten Wörter einen TF-IDF Wert ab 0,01 haben. Legt man sich auf diesen Schwellenwert fest, so stellt dies eine strenge Wahl des Schwellenwertes dar. Streng deswegen, da sich in der Spanne von 0,005 bis 0,01 des TF-IDF Wertes auch relativ relevante Wörter befinden können wie beispielsweise „food“ oder „pharmaceutical“. Möchte man den Grenzwert ein wenig lockern, dann wäre 0,005 der passendere Schwellenwert. Andererseits jedoch würde der Begriff „tool“ auch in die Gruppe der relevanten Wörter aufgenommen, obwohl dieses Wort zu allgemein[264] ist. Aus diesem Grund ist die strenge Auswahl des Schwellenwertes notwendig und empfehlenswert. Eine weitere Herabsetzung des Schwellenwertes unter 0,005 ist ausgeschlossen, da bei näherer Betrachtung der Tabelle, sich darunter Wörter befinden, die eher einen allgemeinen Charakter haben. Weitere Testergebnisse siehe Anhang C.

[258] Schwer Industry wie Metal
[259] Lebensmittel
[260] Konsumgut
[261] Gesundheit
[262] Pharmacie
[263] Naturwissenschaft
[264] Medical tool, Industry tool etc.

9. Der OWL Export

Als letzer Vorgang in der Anwendung erfolgt der OWL Export. Nachdem in der semi-automatischen Kategorisierung die passenden Keywörter extrahiert wurden, gilt es jetzt, die zu kategorisierende Marktstudie in OWL abzulegen. Für den Export wird das *Jena Semantic Web Framework*[265] verwendet. Dieses Framework verfügt über etliche Programmierschnittstellen u. a. für die Abfragesprache *SPARQL*, RDF und die *JENA Ontologie API*. Letztere wird für die Überführung der Marktstudien in OWL benutzt. Alle anderen APIs sind in diesem Fall nicht relevant. Eine alternative API für Ontologien stellt *OWL API*[266] dar. Das *Jena Semantic Web Framework* hat jedoch erhebliche Vorteile gegenüber dem *OWL API*, was im nächsten Kapitel näher geschildert wird.

9.1. Die Jena Ontology API und deren Implementierung

Bei *Jena Semantic Web Framework* handelt es sich um ein Open Source Framework. Die dazugehörige *Jena Ontology API* verwendet man für die Manipulation und Speicherung von Ontologien. In der Literatur[267] [268] wird für die Arbeit mit OWL diese API bzw. dieses Framework empfohlen. Die Tatsache, dass das Framework im Sinne von Open Source frei zur Verfügung steht, vestärkt seine Bevorzugung. Ein Vorteil gegenüber der *OWL API* besteht in der Möglichkeit, OWL Sprachtypen[269] anzugeben. Aus den *OWL API* Dokumentationen[270] ist nicht ersichtlich wie die Angabe der OWL Sprachtypen von statten gehen soll. Bei der *Jena Ontology API*[271] ist dies über das Objekt *ModelFactory* und den dazugehörige Operator *createOntologyModel* möglich. Diesem Operator übergibt man den OWL Sprachtyp als Parameter in Form von Konstanten, die als Modelspezifikation bezeichnet werden. Diese sogenante Modelspezifikation wird über das Objekt *OntModelSpec* repräsentiert.

```
2 /*auswahl von OWL DL als Sprachtyp*/
3 SOURCE=URL der OWL Datei;
4 OntModel base = ModelFactory.createOntologyModel( OntModelSpec.OWL_DL_MEM );
```

[265] http://jena.sourceforge.net
[266] http://owlapi.sourceforge.net
[267] [SEBT09], S. 206
[268] [HFBL09], S. 269
[269] s. S. 37 K. 3.3.1
[270] http://owlapi.sourceforge.net/documentation.hml
[271] s. [DIC09]

```
5 base.read( SOURCE, "RDF/XML" );
```

Listing 29: Grundsätzliche Erstellung einer Ontologie über Jena API

Darüber hinaus ist es bei *Jena Ontology API* möglich, die bevorzugte Serialisierungs-Syntax zu definieren[272] [273], die über den Operator *read* angegeben wird. Gleichzeitig liest *read* auch den OWL Quellcode aus. Als Sprachtyp wird wie in Kapitel 4.2 erwähnt die OWL DL verwendet.

9.1.1. Grundlegende Klassen und Schnittstellen

Vor der Erstellung der Klassen und Schnittstellen wird zunächst auf Basis der in Kapitel 4.8 konzipierten Marktstudienontologie eine OWL Datei erstellt. Diese Ontologie ist Dreh- und Angelpunkt für die Erzeugung von Marktstudieninstanzen[274]. Folgendes Listing zeigt in Auszügen die OWL Datei zur Marktstudienontologie[275].

```
2 <?xml version="1.0"?>
3 ....
4 .....
5 <!-- Angabe der Kategorien Ontologien und der Publisher Ontologie über
6 Namespaces-->

8 <rdf:RDF xmlns="http://localhost/xampp/ontologies/market_report.owl#"
9 ...
10 ....
11  xmlns:categories_onto="http://localhost/xampp/ontologies/categories_onto.owl#"
12  .....
13  xmlns:report_publisher="http://localhost/xampp/ontologies/publisher.owl#"
14 >
15  ............

17  <owl:ObjectProperty rdf:about="#belongsToCategory">
18  <rdfs:domain rdf:resource="#report"/>
19  <rdfs:range rdf:resource="&categories_onto;market_categories"/>
20  </owl:ObjectProperty>

22  <owl:ObjectProperty rdf:about="#publishedBy">
23  <rdfs:domain rdf:resource="#report"/>
```

[272] s. S. 34 K. 3.2.3
[273] s. S. 37 K. 3.3
[274] s. S. 63 K. 4.9
[275] Den kompletten Quellcode finden Sie in Anhang B

```
24 <rdfs:range rdf:resource="&report_publisher;publisher"/>
25 </owl:ObjectProperty>

27 <!-- Name der Marktstudien Klasse-->
28 <owl:Class rdf:about="#report">
29 <rdfs:subClassOf>
30  <owl:Restriction>
31    <owl:onProperty rdf:resource="#belongsToCategory"/>
32    <owl:someValuesFrom rdf:resource="&categories_onto;market_categories"/>
33 </owl:Restriction>
34 </rdfs:subClassOf>
35 </owl:Class>

37 </rdf:RDF>
```

Listing 30: Auszug aus der Marktstudien OWL

Wichtig hierbei ist die definierte Referenz zur Kategorienontologie im OWL Header, die über ein *Namespace* deklariert wird. Hinter der Definition des *Namespace* folgt die URL zur Kategorienontologie. Desweiteren wird die Marktstudienklasse selbst deklariert, die den Namen „report“ trägt. Für eine fehlerfreie Kategorisierung wird die *Property* „belongsToCategory“ in Bezug auf die Marktstudienklasse mit einer zusätzlichen Einschränkung versehen. Auf diese *Property* muss bei der späteren Zuordnung mindestens eine Kategorieangabe folgen. Dafür wird das im Kapitel 4.9 vorgestellte Sprachelement *owl:someValuesFrom* verwendet. Über diesen Weg ordnet man letztendlich die Marktstudien den Kategorien zu und mit Hilfe von *owl:someValuesFrom* wird eine zusätzliche Bedingung definiert, die besagt, dass eine Marktstudie mindestens einer Kategorie zugeordnet werden muss.

Verantwortlich für die Erzeugung von Instanzen auf Basis dieser OWL Datei ist die Schnittstelle *IOWLExport*.

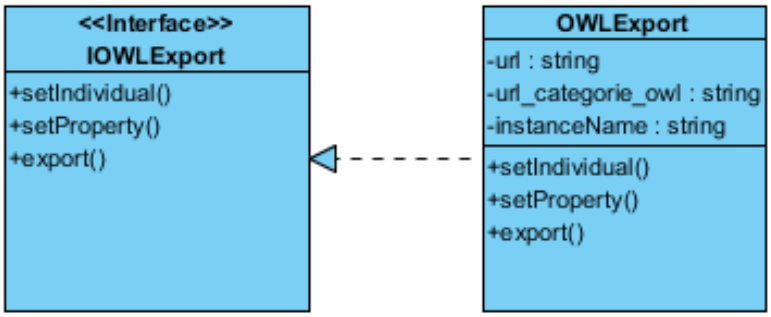


Abbildung 9.1: Die Schnittstelle *IOWLExport* und die zugehörigen Realisierungsklassen

IOWLExport erfüllt folgende grundlegende Aufgaben:

- Instanz erzeugen über *setIndividual()*.
- Über *setProperties* werden Sowohl *DatatypeProperty* also auch *ObjectProperty* definiert[276].
- Export bzw. die Speicherung der zu kategorisierenden Marktstudie in eine OWL Datei.

Allerdings wird keine neue Datei erzeugt, sondern man greift auf die OWL Datei der Marktstudienontologie zurück und erzeugt dort die Instanzen. Der in der Schnittstelle aufgeführte Operator *export()* ruft den Operator *write()* des ModelFactory Objekts aus und dieser verhindert eine vollständige Überschreibung der OWL Datei, wenn neue Instanzen erzeugt werden. D. h. bei Neuinstanziierung wird lediglich die OWL Datei mit den neuen Instanzen erweitert. Beim Erzeugen einer Instanz wird die Kategorienzuordnung vorgenommen und somit wurde die in Kapitel 4.9 gestellte Frage, auf welcher Basis die Zuordnung erfolgen soll, beantwortet.

```
<?xml version="1.0"?>
....
<!-- Angabe der Kategorien Ontologien und der Publisher Ontologie über
Namespaces-->
<rdf:RDF xmlns="http://localhost/xampp/ontologies/market_report.owl#"
...
....
 xmlns:categories_onto="http://localhost/xampp/ontologies/categories_onto.owl#"
 .....
 xmlns:report_publisher="http://localhost/xampp/ontologies/publisher.owl#"
>
......
<report rdf:about="BRIC_Diabetes_Drugs_Market">
<hasTitle rdf:dataType="&xsd;string">BRIC Diabetes Drugs Market</hasTitle>
<hasDescription rdf:dataType="&xsd;string">This report contains key facts about the
    diabetes market in BRIC Countries</hasDescription>
<belongsToCategory rdf:resource="&categories_onto;#Pharmaceutical"></belongsToCategory>
<belongsToCategory
rdf:resource="&categories_onto;#Healthcare"></belongsToCategory>
<publishedBy rdf:resource="&report_publisher;#Datamonitor"></publishedBy>
.............
</report>
```

[276] vgl. S. 60 K. 4.8

```
23 </rdf:RDF>
```

Listing 31: Erzeugung eines Marktstudienindividuums

9.2. Die Weiterverwendung der Ontologie

Schon lange versuchte das Unternehmen dytec GmbH, alle vorhandenen Martkstudiendaten in eine Wissensdatenbank zu überführen, um ein erfolgreiches Wissensmanagement zu betreiben und die Masse an Daten besser kontrollieren zu können. Mit diesem OWL Export ist der Schritt zur Zielsetzung bereits getan, denn die so erzeugte Marktstudienontologie dient zu allererst als eine in OWL abgelegte Wissenrepräsentation und wird später als Index einer auf OWL basierenden Suchmaschine zur Verfügung gestellt, z. B. Swoogle[277]. Ein anderer Verwendungszweck für die Marktstudienontologie ist die Übergabe der Daten an sogenannte Reasoner[278] Systeme. Solche Systeme sind Anwendungen, die in der Lage sind, aus in einer Ontologoie hinterlegten Informationen logische Schlussfolgerungen zu ziehen. I. d. R. zielen Reasoner Systeme darauf ab, neue Fakten zu entdecken und abzuleiten[279]. Das *Jena Framework* selbst bietet über die *Inference API* eine Schnittstelle zur Erstellung eines Reasoner Systems an. Ein anderes gängiges Reasoner System ist das Programm *pellet*[280]. Dieses Programm besitzt die Fähigkeit, aus auf OWL DL basierenden Ontologien Schlussfolgerungen zu ziehen[281]. Dies ist u. a. auch der Grund, warum es wichtig ist, einen OWL Sprachtyp auszuwählen, denn falls ein OWL Dokument als OWL DL deklariert wird, wird es auch von Reasoner Systemen oder von semantischen Suchmaschinen als solches behandelt.

[277] http://swoogle.umbc.edu/
[278] engl. für Schlusfolgern
[279] s. S. 17 K. 2
[280] `http://clarkparsia.com/pellet/`
[281] [HFBL09], S. 36

10. Die Quelldatei zu den Marktstudien

Um die *Text Extraction* und die nachfolgenden Vorgänge erfolgreich durchzuführen, bedarf es einer Quelldatei, in der die Daten zu den Marktstudien hinterlegt sind, die dann eingelesen werden müssen. Wie in der Einführung[282] schon erwähnt ist die gebräuchlichste Quelldatei eine CSV Datei, da die meisten Kunden in dieser Form ihre Daten zur Verfügung stellen. Einige wenige Kunden stellen ihre Daten auch als XML zur Verfügung, dies birgt den Vorteil, dass auf diese Daten auch direkt über eine URL zugegriffen werden kann. Aus diesem Grund wurde die Anwendung so entwickelt, dass sowohl CSV als auch XML Quelldateien eingelesen werden können. Ein vollständiges Ersetzen der CSV Form durch XML kam aus Kundensicht nicht in Frage, da einige Unternehmen kein Interesse hatten, von CSV auf XML umzusteigen. Nichts desto trotz erlangt die Anwendung durch die Fähigkeit, zwei Quelldateiarten lesen zu können, einen hohen Grad an Flexibilität. V. a. durch das XML Format kann jegliche Art von Daten, seien es Martkstudien, generelle Nachrichten oder Pressemitteilungen[283], gelesen werden. Beim Einlesen der XML Datei kommt die *Java SAX API*[284] [285] zum Einsatz. Auf der anderen Seite verwendet man für das Einlesen der CSV Datei die herkömmliche *FileReader*[286] Klasse von Java.

[282] s. S. 15 K. 1.4
[283] vgl. S. 77 K. 6.1
[284] Simple API for XML
[285] `http://www.saxproject.org/`
[286] `http://java.sun.com/j2se/1.4.2/docs/api/java/io/FileReader.html`

11. Fazit & Ausblick

Bei der Entwicklung der Anwendung wurde eigens eine Java GUI[287] gestaltet, jedoch dient sie lediglich zu Präsentationszwecken und um eine bessere Übersicht zu erhalten. Ziel war es, zum einen Möglichkeiten und Ansätze zu finden, wie man Marktstudien mit Hilfe von Java automatisch kategorisieren kann. Zum anderen galt es, Mittel und Wege zu finden, die Daten bzw. die zu kategorisierenden Marktstudien in OWL zu überführen. Aus diesem Grunde wurden die Algorithmen zur Lösung der Problematik so entwickelt, dass sie flexibel eingesetzt werden können, was bedeutet, dass der Anwendungskern nicht an eine bestimmte Benutzeroberfläche gebunden ist. Der Programmkern hat den Charakter einer **Programmbibliothek**, die plattformunabhängig ist und zur Lösung der Problematik beiträgt. Folgende Kriterien muss die Benutzeroberfläche erfüllen: Es muss möglich sein, den Schwellenwert manuell einzugeben bzw. zu manipulieren, damit letztendlich der Benutzer noch die Kontrolle über die Programmergebnisse bekommt und die Quelldatei anzugeben, aus der die Daten zu den Marktstudien eingelesen werden[288]. Als Benutzeroberfläche könnte man weiterhin die in der Entwicklung erstellte GUI anwenden. Da die Anwendung auf Java basiert, ist der Einsatz der GUI in allen Plattformen als ausführbares Programm möglich, jedoch wäre dies nur eine Stand-Alone Anwendung, die auf jedem Arbeitsplatz installiert werden müsste. Ein anderes Einsatzgebiet der Programmbibliothek ist z. B. eine Webanwendung, auf die dann mit Hilfe eines Web-Browsers von überall zugegriffen werden kann. Eine Variante für eine Webanwendung ist die Verwendung eines Open Source Web Framework wie beispielsweise *Struts*[289] sowie die Integration der Programmbibliothek in dieses Web Framework. Die so erstellte Webanwendung würde dann als Zusatz dem aktuellen Martkstudienportal dienen und speziell für die Text und Keyword Analysen eingesetzt.

Um die Flexibilität weiter zu erhöhen, werden die Marktstudiendaten aus der Datenbank zusätzlich als XML hinterlegt. Die Algorithmen zur Textanalyse in der Programmbibliothek benutzen die Java SAX[290] API für die serielle Verarbeitung von XML. Die SAX API hat den Vorteil, dass sie XML Dateien schneller und speicherschonender verarbeiten[291]. Ein eventueller Nachteil, den die Programmbibliothek mit sich bringt liegt in der Tatsache, dass die Ergebniswerte der statistischen Berechnungen sowohl bei der relativen Häufigkeit als auch bei TF-IDF nicht exakt sind. Die errechneten Werte sind eher Näherungs-

[287] Graphical User Interface
[288] s. S. 114 K. 10
[289] `http://struts.apache.org/`
[290] Simple API for XML
[291] `http://openbook.galileocomputing.de/javainsel8/javainsel_15_005.htm`

werte, auf deren Basis dann beispielsweise Schwellenwerte bestimmt werden können. Aber wie in Kapitel 8 schon beschrieben, besteht die Hauptaufgabe der Kernbibliothek darin, dem Benutzer bei der Kategorisierung Vorschläge zu unterbreiten. Dadurch stellt die fehlende Exaktheit bei der statistischen Berechnung keine gravierende Schwäche der Programmbibliothek dar.

Für das Untermehmen stellen die Lösungen in der Programmbibliothek auch eine Art Firmen Know-How dar. Dies ist ein Aspekt, der nicht ausser Acht gelassen werden darf und kann. Durch die Flexibilität[292] [293] der Programmbibliothek könnte man andere Textdaten analysieren, die eine gänzlich andere Thematik als Marktstudien beinhalten. Durch die in dieser Arbeit entwickelte Programmbibliothek verfügt die dytec GmbH nun über folgendes Know-How:

- Know-How für eine automatische Zusammenfassung[294]
- Know-How für eine automatische Kategorisierung auf Basis von statistischen Berechnungen
- Know-How für Hinterlegung der Marktstudien in OWL und dadurch Erfahrung im Semantic Web

Eventuell kann man die Programmbibliothek anderen Unternehmen anbieten, die ebenfalls Probleme haben, einen angemessenen Überblick über die Massen ihrer Daten zu bekommen. Ein anderes Einsatzfeld für die Programmbibliothek wäre z. B. die Analyse von E-Mail Anfragen, die das Unternehmen dytec GmbH täglich in großen Mengen erhält. Die Anfragen könnte man somit zusammenfassen und kategorisieren.

Im Großen und Ganzen ist das Einsatzfeld der Programmbibliothek im Unternehmen selbst vielfältig und flexibel. Primär soll aber die Programmbibliothek folgende elementare Aufgabe erfüllen: Die Unterstützung des Benutzers bei der Kategorisierung von Marktstudien.

[292] s. S. 77 K. 6.1
[293] s. S. 114 K. 10
[294] *Text Extraction*

Literaturverzeichnis

[AH08] ANTONIOU, Grigorius ; HERMELEN, Frank van: *A Semantic Web Primer*. MIT Press, 2008

[Aut08] AUTHORS, Various ; MANI, Inderjeet (Hrsg.) ; MAYBURY, Mark (Hrsg.): *Advance in Automatic Text Summarization*. MIT Press, 2008

[BER01] BERNERS-LEE, TIM ; HENDLER, JAMES ; LASILLA, ORA: *The Semantic Web*. http://www.scientificamerican.com/article.cfm?id=the-semantic-web, 2001. – Internet, abgerufen im Februar 2010

[DIC09] DICKINSON, IAN: *The Jena Ontology API*. http://jena.sourceforge.net/ontology/index.html, 2009. – Internet, abgerufen im Mai 2010

[Edm68] EDMUNDSON, H.P.: New Methods in Automatic Extraction. In: MANI, Inderjeet (Hrsg.) ; MAYBURY, Mark (Hrsg.): *Advance in Automatic Text Summarization*. London : MIT Press, 1968, S. 23–42

[EN98] ENDERS-NIGGEMEYER, Brigitte: *Summarizing Information*. Springer, 1998

[Erl01] ERLENKÖTTER, Helmut: *XML Extensible Markup Language von Anfang an*. Rowohlt, 2001

[GRU92] GRUBER, TOM: *What is an Ontology?* http://www-ksl.stanford.edu/kst/what-is-an-ontology.html, 1992. – Internet, abgerufen im Januar 2010

[HFBL09] HEBELER, John ; FISCHER, Matthew ; BLACE, Ryan ; LOPEZ, Andrew P.: *Semantic Web Programming*. Wiley, 2009

[HKRS08] HITZLER, Pascal ; KROETZSCH, Markus ; RUDOLPH, Sebastian ; SURE, York: *Semantic Web*. Springer, 2008

[HQW08] HEYER, Gerhard ; QUASTHOF, Uwe ; WITTIG, Thomas: *Text Mining - Wissensrohstoff Text*. W3L, 2008

[HUG95] HUGHES, KEVIN: *Entering the World-Wide Web: A Guide to Cyberspace*. http://www.faqs.org/faqs/www/guide/, 1995. – Internet, abgerufen im Februar 2010

[Lec08] LECHNER: *Lecnhers Fremdwörterbuch*. Lechner, 2008

[Luh58] LUHN, H.P.: *The Automatic Creation of Literature Abstracts*. http://www.di.ubi.pt/~jpaulo/competence/general/%281958%29Luhn.pdf, 1958. – Internet, abgerufen im April 2010

[Man98] MANI, Inderjeet: *Automatic Summarization*. John Benjamins Publishing Company, 1998

[NOY08] NOY, NATALYA F. ; MCGUINESS, DEBORAH L.: *Ontology Development 101: A Guide to Creating Your First Ontology*. http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html, 2008. – Internet, abgerufen im Februar 2010

[RAN01] RAND, REBECCA ; HILBER, ROSINA : *Implizites Wissen - thetacit dimension*. http://www.uibk.ac.at/psychologie/mitarbeiter/leidlmair/implizites_wissen_ss2006.pdf, 2001. – Internet, abgerufen im Februar 2010

[San10] SANCHEZ, Manuel B.: *Persönliches Interview mit dem Geschäftsführer der dynamic Technologies GmbH geführt vom Verfasser*. Köln, 2010. – Interview am 01.02.2010

[SCH08] SCHULZ, URSULA: *Die Nutzung der Worthäufigkeit zur Ermittlung geeigneter Indexate*. http://www.bui.haw-hamburg.de/pers/ursula.schulz/astep/le6_step_3.html, 2008. – Internet, abgerufen im Mai 2010

[SEBT09] SEGARAN, Toby ; EVANS, Colin ; BLACE, Ryan ; TAYLOR, Jamie: *Programming the Semantic Web*. O'Reilly, 2009

[SM83] SALTON, Gerard ; MCGILL, Michael: *Information Retrieval - Grundlegendes für Informationswissenschaftler*. McGraw-Hill-Texte, 1983

[WAC08] WACH, ELMAR P.: *Semantic E-Commerce*. http://www.e-commerce-magazin.de/de/artikel/do/detail/id/12.html, 2008. – Internet, abgerufen im März 2010

Anhang

Anhang A Beispielanwendung und Systemkomponenten

A.1 Beispielanwendung

Während der Entwicklung wurden vornehmlich Testdaten von reports-research.com übernommen. Für die Testzusammenfassungen und Kategorisierungen konzentrierte man sich zuerst auf eine Marktstudie.

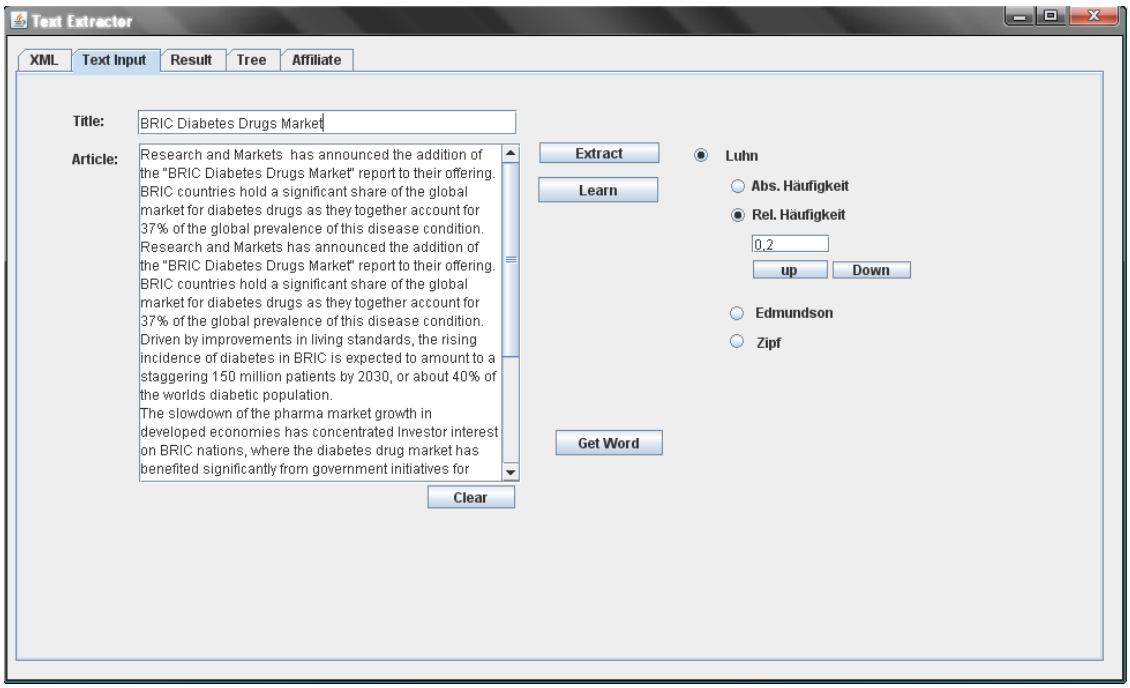


Abbildung A.1: Eingabe einer Marktstudie und Auswahl einer Extraktionsmethode

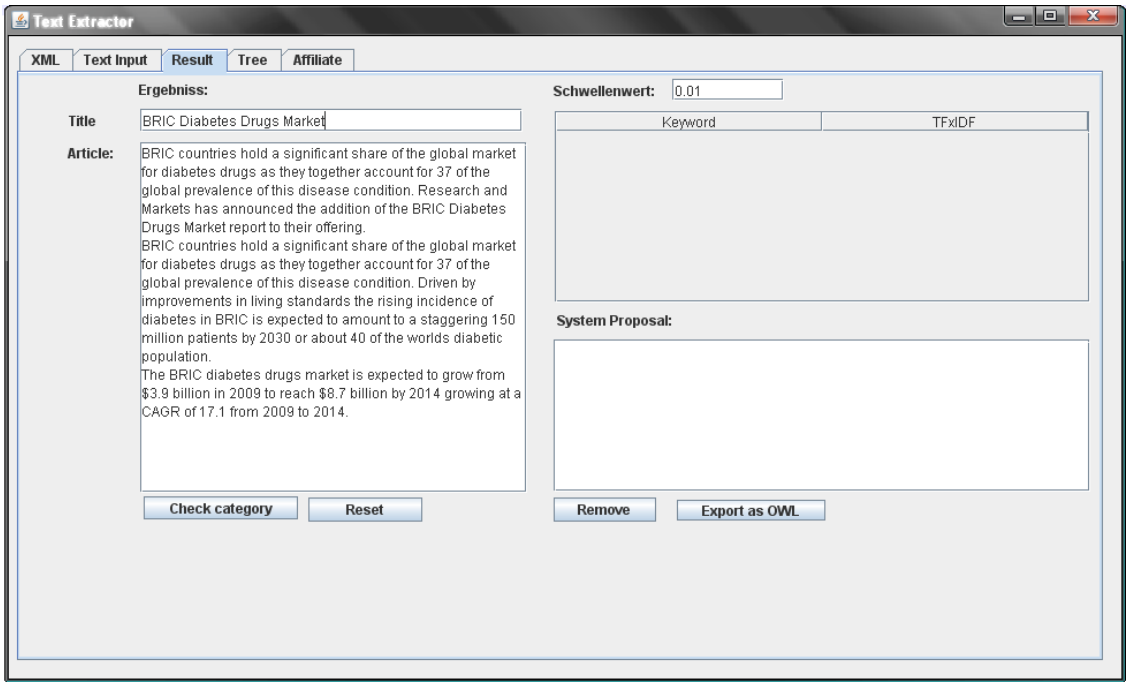


Abbildung A.2: Die Zusammenfassung des Eingabetextes

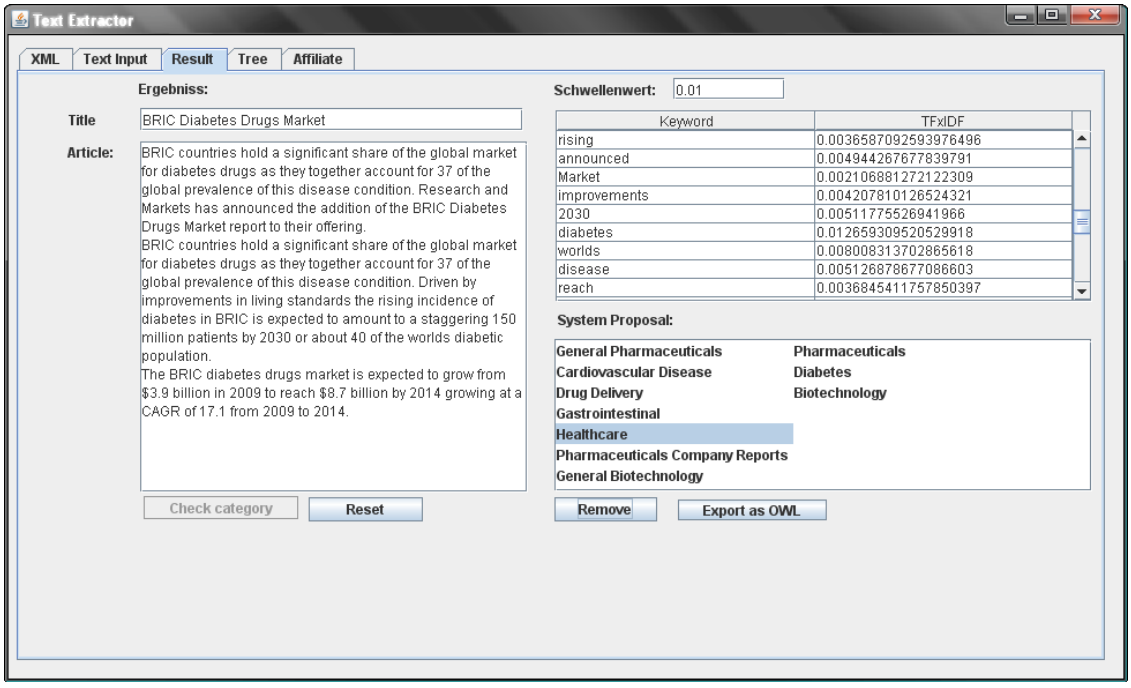


Abbildung A.3: Das System schlägt auf Basis des *Abstract* Kategorien vor

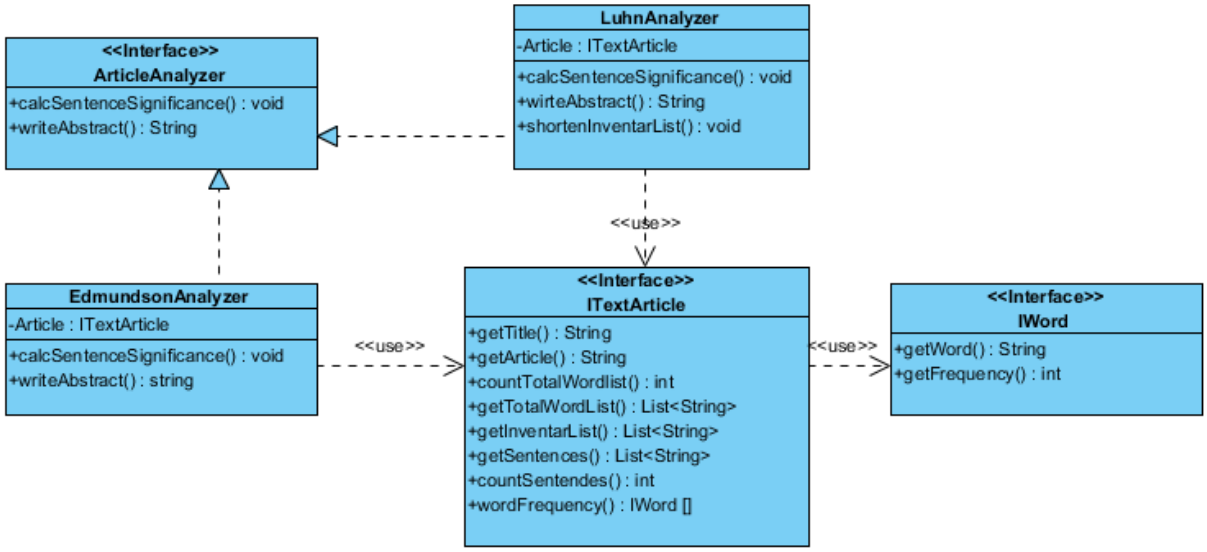


Abbildung A.4: Interfaces und Klassen zu den *Article Analyzern* und die Schnittstelle zum Artikel selbst

A.2 Systemkomponenten und Datenbanken

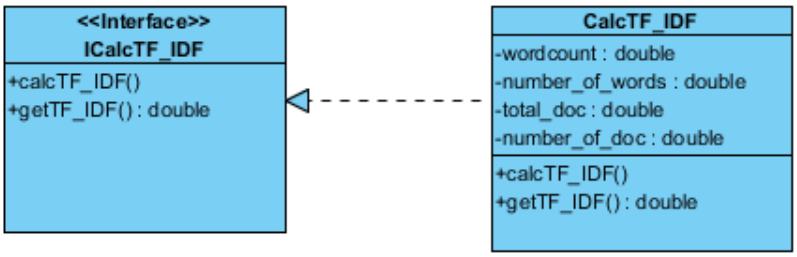


Abbildung A.5: Die Schnittstelle *ICalcTF_IDF* verantwortlich für TF-IDF Berchnungen

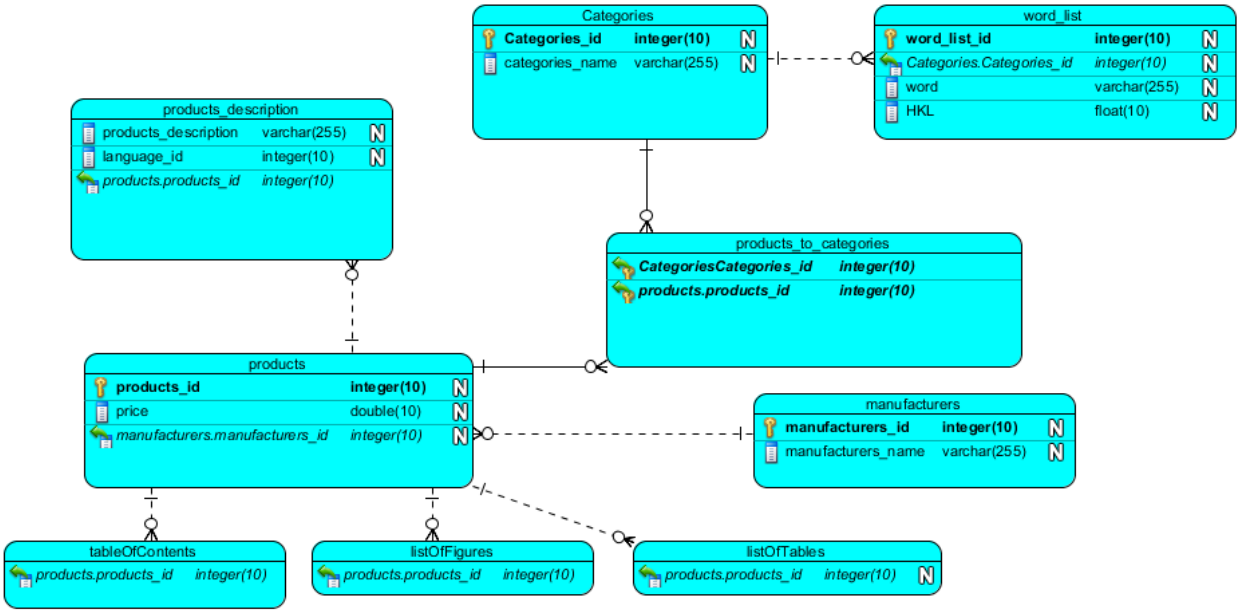


Abbildung A.6: Die Datenbankstruktur mit der neu hinzugekommenen Wortlistentabelle

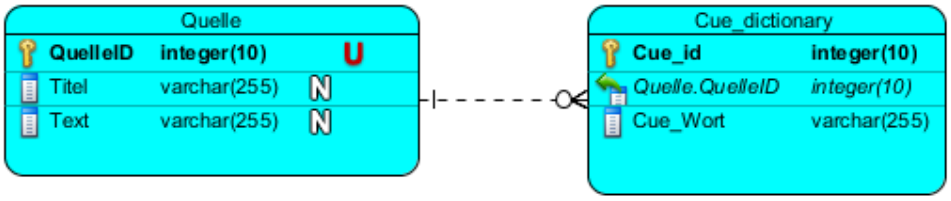


Abbildung A.7: Datenbankstruktur zum *Cue Dictionary*

Anhang B Ontologien

B.1 Die Marktstudienontologie

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
 <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
 <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
 <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
 <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
 <!ENTITY market_report "http://localhost/xampp/ontologies/market_report.owl#" >
 <!ENTITY categories_onto "http://localhost/xampp/ontologies/market_cateogories.owl#" >
 <!ENTITY report_publisher "http://localhost/xampp/ontologies/publisher.owl#" >
]>



<rdf:RDF xmlns="http://localhost/xampp/ontologies/market_report.owl#"
 xml:base="http://localhost/xampp/ontologies/market_report.owl"
   xmlns:categories_onto="http://localhost/xampp/ontologies/market_categories.owl#"
   xmlns:report_publisher="http://localhost/xampp/ontologies/publisher.owl#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:owl="http://www.w3.org/2002/07/owl#"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:market_report="http://localhost/xampp/ontologies/market_report.owl#">

   <owl:DatatypeProperty rdf:about="#hasTitle">
 <rdfs:domain rdf:resource="#report"/>
 <rdfs:range rdf:resource="&xsd;string"/>
 </owl:DatatypeProperty>

  <owl:DatatypeProperty rdf:about="#hasDescription">
 <rdfs:domain rdf:resource="#report"/>
 <rdfs:range rdf:resource="&xsd;string"/>
 </owl:DatatypeProperty>

  <owl:DatatypeProperty rdf:about="#costs">
 <rdfs:domain rdf:resource="#report"/>
 <rdfs:range rdf:resource="&xsd;float"/>
 </owl:DatatypeProperty>
```

```
41  <owl:DatatypeProperty rdf:about="#writtenIn">
42 <rdfs:domain rdf:resource="#report"/>
43 <rdfs:range rdf:resource="&xsd;string"/>
44 </owl:DatatypeProperty>

46  <owl:DatatypeProperty rdf:about="#priceIn">
47 <rdfs:domain rdf:resource="#report"/>
48 <rdfs:range rdf:resource="&xsd;string"/>
49 </owl:DatatypeProperty>

51  <owl:DatatypeProperty rdf:about="#fromYear">
52 <rdfs:domain rdf:resource="#report"/>
53 <rdfs:range rdf:resource="&xsd;gYearMonth"/>
54 </owl:DatatypeProperty>

56  <owl:DatatypeProperty rdf:about="#hasTableOfContents">
57 <rdfs:domain rdf:resource="#report"/>
58 <rdfs:range rdf:resource="&xsd;string"/>
59 </owl:DatatypeProperty>

61  <owl:DatatypeProperty rdf:about="#hasListOfFigures">
62 <rdfs:domain rdf:resource="#report"/>
63 <rdfs:range rdf:resource="&xsd;string"/>
64 </owl:DatatypeProperty>

66  <owl:DatatypeProperty rdf:about="#hasListOfTables">
67 <rdfs:domain rdf:resource="#report"/>
68 <rdfs:range rdf:resource="&xsd;string"/>
69 </owl:DatatypeProperty>

71 <owl:ObjectProperty rdf:about="#belongsToCategory">
72 <rdfs:domain rdf:resource="#report"/>
73 <rdfs:range rdf:resource="&categories_onto;market_categories"/>
74 </owl:ObjectProperty>

76  <owl:ObjectProperty rdf:about="#publishedBy">
77 <rdfs:domain rdf:resource="#report"/>
78 <rdfs:range rdf:resource="&report_publisher;publisher"/>
79 </owl:ObjectProperty>

81 <owl:Class rdf:about="#report">
82 <rdfs:subClassOf>
83  <owl:Restriction>
```

```
84      <owl:onProperty rdf:resource="#belongsToCategory"/>
85      <owl:someValuesFrom rdf:resource="&categories_onto;market_categories"/>
86   </owl:Restriction>
87  </rdfs:subClassOf>
88  </owl:Class>

90  <owl:ObjectProperty rdf:about="#publishedOn">
91   <rdfs:domain rdf:resource="#report"/>
92   <rdfs:range rdf:resource="#Web-Portal"/>
93  </owl:ObjectProperty>

95  <!-- zu jedem Portal existiert eine OWL Klasse-->
96  <owl:Class rdf:about="Web-portal">
97  <owl:intersectionOf rdf:parseType="Collection">
98  <owl:Class rdf:about="www.markt-studie.de" />
99  <owl:Class rdf:about="www.reports-research.com" />
100 <owl:Class rdf:about="www.estudio-mercado.es" />
101 </owl:intersectionOf>
102 </owl:Class>



106 </report>









112 </rdf:RDF>


115 <!-- Generated by the OWL API (version 3.0.0.1413) http://owlapi.sourceforge.net --
        >
```

Listing 32: Komplette Marktstudienontologie

Folgende Abbildung zeigt die Erzeugung eines Marktstudienindividuums

```
2  <report rdf:about="BRIC_Diabetes_Drugs_Market">
3  <hasTitle rdf:dataType="&xsd;string">BRIC Diabetes Drugs Market</hasTitle>
4  <hasDescription rdf:dataType="&xsd;string">This report contains key facts about the
       diabetes market in BRIC Countries</hasDescription>
5  <belongsToCategory rdf:resource="http://localhost/xampp/ontologies/market_cateogories.owl
       #Pharmaceutical"></belongsToCategory>
```

```
<belongsToCategory rdf:resource="http://localhost/xampp/ontologies/market_cateogories.owl
    #Healthcare"></belongsToCategory>
<publishedBy rdf:resource="http://localhost/xampp/ontologies/publisher.owl#Datamonitor"><
    /publishedBy>
<publishedOn rdf:resource="#www.markt-studie.de"></publishedOn>
<costs rdf:dataType="&xsd;float">200.00</costs>
<priceIn rdf:dataType="&xsd;string">USD</priceIn>
<writtenIn rdf:dataType="&xsd;string">english</writtenIn>
<fromYear rdf:dataType="&xsd;gYearMonth">2009/10</fromYear>
<hasTableOfContents rdf:dataType="&xsd;string">Table of Contents.....</hasTableOfContents
    >
<hasListOfFigures rdf:dataType="&xsd;string">List of Figures.....</hasListOfFigures>
<hasListOfTables rdf:dataType="&xsd;string">List of Tables.......</hasListOfTables>
</report>
```

Listing 33: Erzeugung des Marktstudien Individuums

In dem OWL Tool *Protege* werden Individuen in Normaltext ausgeschrieben, um eine bessere Übersicht über einzelne *Statements* und ihre zugehörigen Individuen in der Marktstudienontologie zu gewährleisten.

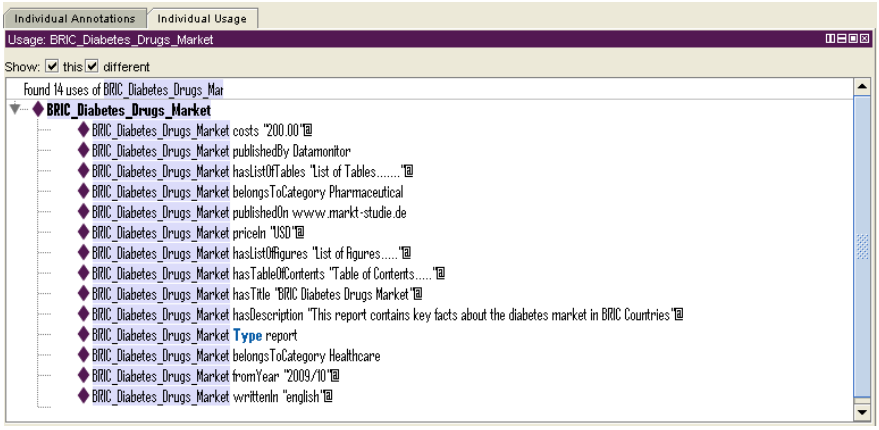


Abbildung B.1: Ausgabe des Individuums als Text in *Protege*

B.2 Die Publisherontologie

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
 <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
```

```
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
<!ENTITY publisher "http://localhost/xampp/ontologies/publisher.owl#" >
]>


<rdf:RDF xmlns="http://localhost/xampp/ontologies/publisher.owl#"
 xml:base="http://localhost/xampp/ontologies/publisher.owl"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:owl="http://www.w3.org/2002/07/owl#"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:publisher="http://localhost/xampp/ontologies/publisher.owl#">


<owl:Class rdf:about="report_publisher" />

<!-- Marktstudien Hersteller -->

<owl:Class rdf:about="Datamonitor">
 <rdfs:subClassOf>
  <owl:Class rdf:about="report_publisher" />
 </rdfs:subClassOf>
 </owl:Class>

 <owl:Class rdf:about="BBE">
 <rdfs:subClassOf>
  <owl:Class rdf:about="report_publisher" />
 </rdfs:subClassOf>
 </owl:Class>

<owl:Class rdf:about="CCM-International-Limited">
 <rdfs:subClassOf>
  <owl:Class rdf:about="report_publisher" />
 </rdfs:subClassOf>
 </owl:Class>

 <owl:Class rdf:about="Eupd-Research">
 <rdfs:subClassOf>
  <owl:Class rdf:about="report_publisher" />
 </rdfs:subClassOf>
 </owl:Class>
```

```
<owl:Class rdf:about="Visiongain">
 <rdfs:subClassOf>
  <owl:Class rdf:about="report_publisher" />
 </rdfs:subClassOf>
 </owl:Class>

 <owl:Class rdf:about="Globaldata">
 <rdfs:subClassOf>
  <owl:Class rdf:about="report_publisher" />
 </rdfs:subClassOf>
 </owl:Class>

 <owl:Class rdf:about="Ovum">
 <rdfs:subClassOf>
  <owl:Class rdf:about="report_publisher" />
 </rdfs:subClassOf>
 </owl:Class>

 <owl:Class rdf:about="MSI">
 <rdfs:subClassOf>
  <owl:Class rdf:about="report_publisher" />
 </rdfs:subClassOf>
 </owl:Class>

<owl:Class rdf:about="Verdict-Research">
 <rdfs:subClassOf>
  <owl:Class rdf:about="report_publisher" />
 </rdfs:subClassOf>
 </owl:Class>

<owl:Class rdf:about="TriMark-Publications">
 <rdfs:subClassOf>
  <owl:Class rdf:about="report_publisher" />
 </rdfs:subClassOf>
 </owl:Class>

 <owl:Class rdf:about="Zukunftsinstitut-GmbH">
 <rdfs:subClassOf>
  <owl:Class rdf:about="report_publisher" />
 </rdfs:subClassOf>
 </owl:Class>

 <owl:Class rdf:about="GBI-Research">
 <rdfs:subClassOf>
```

```
95   <owl:Class rdf:about="report_publisher" />
96  </rdfs:subClassOf>
97  </owl:Class>

99  <owl:Class rdf:about="Faktenkontor-GmbH">
100 <rdfs:subClassOf>
101  <owl:Class rdf:about="report_publisher" />
102 </rdfs:subClassOf>
103 </owl:Class>

105 </rdf:RDF>
```

Listing 34: Die Publisherontologie

B.3 Die Kategorienontologie

Die komplette Kategorienontologie entnehmen Sie bitte der beigelegten CD.

Anhang C Extraktionsergebnisse

C.1 Extraktionsergebnisse nach Luhn

C.1.1 Text Extraction einer Marktstudie mit 230 Wörtern

Titel: „BRIC Diabetes Drugs Market“[295]
Anzahl Gesamtwörter: 230
Inventarliste: 134
Anzahl Sätze: 5
gekürzte Inventarliste: 7
Schwellenwert für relative Häufigkeit: 0,2
Eingabe:
„Research and Markets has announced the addition of the "BRIC Diabetes Drugs Marketreport to their offering. BRIC countries hold a significant share of the global market for diabetes drugs as they together account for 37Research and Markets has announced the addition of the "BRIC Diabetes Drugs Marketreport to their offering. BRIC countries hold a significant share of the global market for diabetes drugs as they together account for 37The slowdown of the pharma market growth in developed economies has concentrated Investor interest on BRIC nations, where the diabetes drug market has benefited significantly from government initiatives for spreading patient awareness, and the subsequent increase in the uptake of novel drugs such as incretin mimetics (Eli Lillys Byetta) and dipeptidyl inhibitors (Mercks Januvia and Novartis Galvus). The BRIC diabetes drugs market is expected to grow from $3.9 billion in 2009 to reach $8.7 billion by 2014, growing at a CAGR of 17.1% from 2009 to 2014. Oral drugs accounted for around 65% of the market in 2009, but injectables are“
Abstract:
"BRIC countries hold a significant share of the global market for diabetes drugs as they together account for 37 of the global prevalence of this disease condition. Research and Markets has announced the addition of the BRIC Diabetes Drugs Market report to their offering. Driven by improvements in living standards the rising incidence of diabetes in BRIC is expected to amount to a staggering 150 million patients by 2030 or about 40 of the worlds diabetic population. The BRIC diabetes drugs market is expected to grow from $3.9 billion in 2009 to reach $8.7 billion by 2014 growing at a CAGR of 17.1 from 2009 to 2014.“

[295] http://www.encyclopedia.com/doc/1G1-224335984.html

C.1.2 Text Extraction einer Marktstudie mit 439 Wörtern

Titel: „The Global Liquefied Natural Gas (LNG) Market, 2010-2020“[296]
Anzahl Gesamtwörter: 439
Inventarliste: 246
Anzahl Sätze: 19
gekürzte Inventarliste: 4
Schwellenwert für relative Häufigkeit: 0,3
Eingabe:
The Global Liquefied Natural Gas (LNG) Market, 2010-2020 is our new energy report. The report outlines trends in the LNG industry and anticipates the direction of the market over the next decade, outlining the strengths and weaknesses of different markets with targeted sales forecasts.

Based on our research, global spending in 2010 on new LNG infrastructure will total $24bn. We analyse, quantify and forecast the expected spending in the global and regional LNG infrastructure markets over the period 2010-2020. These forecasts are underpinned by Visiongain's forecast of LNG demand as an emerging commodity over the period 2010-2020.

This report offers in depth analysis of LNG as one of the key components of the energy industry - an industry which is undergoing broad shifts. LNG is playing a growing role within the natural gas market and the energy world, but this growth is developing in particular directions which are outlined in this report. The report analyses a wealth of data and introduces a clear analysis of where the market will develop based upon diverse factors and insight into the market, anticipating how and why the market will evolve from 2010 onwards.

The report also describes the most important technological changes within the LNG industry and assesses their importance for the growth of the market over the long term. The various drivers and restraints of the market are assessed in order to provide readers with specific insights into the future direction of the LNG market.

How much are individual regions planning to spend on acquiring new LNG infrastructure and upgrading and maintaining existing infrastructure between 2010 and 2020? How much will LNG demand increase over the period 2010-2020? Who are the leading companies in the LNG industry? Where are the growth opportunities over the next decade in which geographical region and with which type of technology? These critical questions

[296] http://www.reports-research.com/market-surveys/global-liquified-natural-market-20102020-p-75759.html

and many more are definitively answered in this comprehensive report.

Comprehensive analysis of the global LNG market.

The Global Liquefied Natural Gas (LNG) Market, 2010-2020 report examines this sector critically with a comprehensive review of recent contracts, news reports, industry publications, market analysis and expert consultation. The report provides detailed sales forecasts for the global market, regional market forecasts; a strengths, weaknesses, opportunities and threats (SWOT) analysis; discussions of commercial and technological trends; and assessments of market drivers and restraints. This report also includes transcripts of in-depth interviews with industry experts. This package of analyses cannot be obtained anywhere else.

The report draws on a rich combination of primary and secondary research, interviews, official corporate and governmental announcements, media reports, policy documents, industry statements and an extensive consultation of expert opinion.

Abstract:

The Global Liquefied Natural Gas LNG Market 2010-2020 report examines this sector critically with a comprehensive review of recent contracts news reports industry publications market analysis and expert consultation. The report provides detailed sales forecasts for the global market regional market forecasts a strengths weaknesses opportunities and threats SWOT analysis discussions of commercial and technological trends and assessments of market drivers and restraints.

C.1.3 Text Extraction einer Marktstudie mit 55 Wörtern

Titel: „New Types of HIV Drugs“[297]
Anzahl Gesamtwörter: 55
Inventarliste : 22
Anzahl Sätze: 2
gekürzte Inventarliste: 2
Schwellenwert für relative Häufigkeit: 0,1
Eingabe:

This TriMark Publications Database Table is a one-page table of hard-to-find numerical information. The Database Table is derived from a proprietary source and is meant to simply provide a high-level overview of specific data points. The Database Table is NOT a report or a comprehensive analysis. Such studies are offered by TriMark Publications at

[297] http://www.reports-research.com/market-surveys/types-drugs-p-51617.html

http://www.trimarkpublications.com.
Abstract:
This TriMark Publications Database Table is a one-page table of hard-to-find numerical information.

C.1.4 Text Extraction eines Wikipedia Artikels mit 101 Wörtern

Titel: „Drugs“[298]
Anzahl Gesamtwörter: 101
Inventarliste: 44
Anzahl Sätze: 3
gekürzte Inventarliste: 8
Schwellenwert für relative Häufigkeit : 0,1
Eingabe:
Recreational drugs are chemical substances that affect the central nervous system such as opioids or hallucinogens. Drugs are usually distinguished from endogenous biochemicals by being introduced from outside the organism. For example insulin is a hormone that is synthesized in the body it is called a hormone when it is synthesized by the pancreas inside the body but if it is introduced into the body from outside it is called a drug. Many natural substances such as beers wines and some mushrooms blur the line between food and drugs as when ingested they affect the functioning of both mind and body.
Abstract:
For example insulin is a hormone that is synthesized in the body it is called a hormone when it is synthesized by the pancreas inside the body but if it is introduced into the body from outside it is called a drug. Many natural substances such as beers wines and some mushrooms blur the line between food and drugs as when ingested they affect the functioning of both mind and body.

C.2 Extraktionsergebnisse nach Edmundson

C.2.1 Text Extraction einer Marktstudie mit 363 Wörtern

Titel: „Diabetes Market UAE“[299]
Anzahl Gesamtwörter : 363

[298] Erster Absatz aus `http://en.wikipedia.org/wiki/Drug`
[299] `http://www.reports-research.com/market-surveys/diabetes-market-p-69905.html`

Eingabe:
Diabetes is one of the fastest growing lifestyle and debilitating diseases in the Middle East region. At present, one out of every five person in the UAE is suffering from diabetes. The concern becomes a bit serious as diabetes is associated with several other chronic diseases like cardiovascular diseases. This has put an extra burden on the country's healthcare spending to allocate more funds for diagnosis, care and prevention. According to our new research report „Diabetes Market in UAE“ the UAE diabetes care market is projected to grow at a CAGR of more than 8% during 2010-2012.

We have studied the UAE diabetes market by segmenting it into - insulin and non-insulin market. We have found that although insulin care products have a smaller share in the market by volume, they contribute maximum to the total market sales. Each of these two segments have been thoroughly studied and analyzed in the report. Moreover, the report provides an overview of patient profile in the country which will help clients to identify the future demand scenario.

Our report has found that the prevalence of diabetes in the UAE has reached an epidemic level and some serious steps need to be taken by the government to spread awareness among people about its social, economic and personal effects. The report has also found that the demand for blood glucose meters and other medical devices will increase at a rapid pace and will introduce major changes in the current market scenario. We suggest that the government should make some changes in regulatory environment and promote the local manufacturing of drugs and devices.

The report studies the current development process and its effects on various segments of the industry. The research gives detailed statistical and analytical review of macroeconomic indicators, key drivers and restraints. It contains exhaustive information about the market players, government initiatives and patient profile that will help clients to draw up market strategies and assess opportunity areas in the UAE diabetes market.

The report also includes a section of future projection and demand analysis that gives a rough idea about the prospective market trends to facilitate clients comprehend how the market will progress in coming years.

Abstract:
Our report has found that the prevalence of diabetes in the UAE has reached an epidemic level and some serious steps need to be taken by the government to spread awareness among people about its social economic and personal effects. The report also includes a section of future projection and demand analysis that gives a rough idea about the prospective market trends to facilitate clients comprehend how the market will progress in coming years. We have found that although insulin care products have a smaller share

in the market by volume they contribute maximum to the total market sales. The report has also found that the demand for blood glucose meters and other medical devices will increase at a rapid pace and will introduce major changes in the current market scenario. The report studies the current development process and its effects on various segments of the industry. We suggest that the government should make some changes in regulatory environment and promote the local manufacturing of drugs and devices. Moreover the report provides an overview of patient profile in the country which will help clients to identify the future demand scenario. According to our new research report „Diabetes Market in UAE“ the UAE diabetes care market is projected to grow at a CAGR of more than 8 during 2010-2012. We have studied the UAE diabetes market by segmenting it into insulin and non-insulin market. At present one out of every five person in the UAE is suffering from diabetes. The concern becomes a bit serious as diabetes is associated with several other chronic diseases like cardiovascular diseases.

Ausgewählte Sätze:

Satz	Satzsignifikanz
At present one out of every five person in the UAE is suffering from diabetes.	7
The concern becomes a bit serious as diabetes is associated with several other chronic diseases like cardiovascular diseases.	6
According to our new research report Diabetes Market in UAE the UAE diabetes care market is projected to grow at a CAGR of more than 8% during 2010-2012.	28
We have studied the UAE diabetes market by segmenting it into insulin and non-insulin market.	23
We have found that although insulin care products have a smaller share in the market by volume they contribute maximum to the total market sales.	97
Moreover the report provides an overview of patient profile in the country which will help clients to identify the future demand scenario.	29
Our report has found that the prevalence of diabetes in the UAE has reached an epidemic level and some serious steps need to be taken by the government to spread awareness among people about its social economic and personal effects.	140

The report has also found that the demand for blood glucose meters and other medical devices will increase at a rapid pace and will introduce major changes in the current market scenario.	86
We suggest that the government should make some changes in regulatory environment and promote the local manufacturing of drugs and devices.	55
The report studies the current development process and its effects on various segments of the industry.	58

C.2.2 Text Extraction einer Marktstudie mit 251 Wörtern

Titel: „Production and Market of Threonine in China“[300]
Anzahl Gesamtwörter : 251
Eingabe:
The commercial production of threonine, one of the amino acids, started from the 1990s. About 19 kinds of amino acids are being produced in China now, and the production scale of the manufacturers is very small. The Chinese manufacturers mainly focus on pharma-grade threonine.

China is a market with great potential for amino acid. According to the statistics in mid 2001, there were over 100 amino acids factories in China. The actual production is estimated to be more than 500,000t/a in China, but about 90% 95% is glutamic acid. There is a lot space for other amino acids. The central government has already implemented many policies to guide the direction of amino acids.

China is a promising market for threonine. But at present China's market demand for threonine is only about 100t/a.

The small market volume is mainly because Chinese producers only aim at the market of amino acid transfusion. In China, the amino acid transfusion industry developed fast in the previous years. In 1989, there were only 8 manufacturers producing amino acid transfusion, but the number of manufacturers climbed up to 80 in 1999.

The feed grade threonine relies on import. Now promoted by the rapid growth of livestock industry and pharmaceuticals, the production of threonine will be more flourishing, and China's import of threonine is likely to decrease in the future.

[300] http://www.reports-research.com/market-surveys/production-market-threonine-china-p-22259.html

In this report, comprehensive tel interview with experts within the industry and further analysis in detail are carried out to verify the up-to-date situation of the market.

Abstract:

The small market volume is mainly because Chinese producers only aim at the market of amino acid transfusion. In China the amino acid transfusion industry developed fast in the previous years. Now promoted by the rapid growth of livestock industry and pharmaceuticals the production of threonine will be more flourishing and Chinas import of threonine is likely to decrease in the future. China is a market with great potential for amino acid. About 19 kinds of amino acids are being produced in China now and the production scale of the manufacturers is very small. In this report comprehensive tel interview with experts within the industry and further analysis in detail are carried out to verify the up-to-date situation of the market. In 1989 there were only 8 manufacturers producing amino acid transfusion but the number of manufacturers climbed up to 80 in 1999. China is a promising market for threonine. The commercial production of threonine one of the amino acids started from the 1990s. The actual production is estimated to be more than 500000t/a in China but about 90 95 is glutamic acid. But at present Chinas market demand for threonine is only about 100t/a. The central government has already implemented many policies to guide the direction of amino acids.

Ausgewählte Sätze:

Satz	Satzsignifikanz
The commercial production of threonine one of the amino acids started from the 1990s.	19
About 19 kinds of amino acids are being produced in China now and the production scale of the manufacturers is very small.	44
China is a market with great potential for amino acid.	46
The actual production is estimated to be more than 500000t/a in China but about 90 95 is glutamic acid.	8
The central government has already implemented many policies to guide the direction of amino acids.	5
China is a promising market for threonine.	23
But at present Chinas market demand for threonine is only about 100t/a.	6

The small market volume is mainly because Chinese producers only aim at the market of amino acid transfusion.	53
In China the amino acid transfusion industry developed fast in the previous years.	49
In 1989 there were only 8 manufacturers producing amino acid transfusion but the number of manufacturers climbed up to 80 in 1999.	24
Now promoted by the rapid growth of livestock industry and pharmaceuticals the production of threonine will be more flourishing and Chinas import of threonine is likely to decrease in the future.	48
In this report comprehensive tel interview with experts within the industry and further analysis in detail are carried out to verify the up-to-date situation of the market.	36

Anhang D Veränderungen der TF-IDF Werte

Die folgenden Tabellen zeigen, wie sich die einzelnen TF-IDF Werte je Wort bei steigender Anzahl des Gesamtkorpus verändern.

D.1 TF-IDF Werte in Relation zu 400 Dokumenten im Gesamtkorpus

Wort	TF-IDF
epigenetic	0,06437698636167312
milk	0,03202070199995807
bioinformatics	0,021415664870860104
livestock	0,020137091127160806
insulin	0,014710569796119455
metal	0,014240733956509757
beer	0,011721653089045601
diabetes	0,011460997630065688
food	0,009582538284157309
tools	0,008031373204019317
body	0,005515107429889175
pharmaceutical	0,0052548919555207695
introduced	0,003708735217884334
introduction	0,0036207770268922566
public	0,003415592477652624
line	0,002916571062106434
forecast	0,0029079965737533035
report	0,002437779400987902
market	0,00199159224426669
inside	0,0016078599347192515
year	0,0013994780608796635
sector	0,0013527208793540212
revenue	7,273746357565604E-4
key	2,8869849374389933E-4

D.2 TF-IDF Werte in Relation zu 500 Dokumenten im Gesamtkorpus

Wort	TF-IDF
epigenetic	0,0665095660896914
milk	0,033320466472331584
bioinformatics	0,025133508990323048
livestock	0,02095448343867433
insulin	0,01530769211996457
metal	0,014914593135948102
beer	0,012109949608800737
diabetes	0,012111066760951259
food	0,009881114441962772
tools	0,008475829141129142
body	0,006971656039843916
pharmaceutical	0,005887787558712145
introduced	0,0038687183955446775
introduction	0,0037769659647189473
public	0,003577215351257018
line	0,003059862404705957
forecast	0,002420136875520021
report	0,00205127766145591
market	0,001957231645558062
inside	0,0017773188516890255
year	0,0017137652970933667
sector	0,0015604904176907898
revenue	9,678328309094301E-4
key	3,496893141852698E-4

D.3 TF-IDF Werte in Relation zu 600 Dokumenten im Gesamtkorpus

Wort	TF-IDF
epigenetic	0,06830810002948565
milk	0,034416636752199546
bioinformatics	0,025899152487932644

Wort	TF-IDF
livestock	0,021643839993587315
insulin	0,01581128162310696
metal	0,015482899487611676
beer	0,012437423634886768
diabetes	0,012659309520529918
food	0,010110880158720862
tools	0,00732252702097262
body	0,0046421053876906935
pharmaceutical	0,006421546839892224
introduced	0,004003641920812191
introduction	0,0039086895772109205
public	0,002899257749524025
line	0,0031807086922130664
forecast	0,002080582106031727
report	0,0017032488045080323
market	0,001815348306543681
inside	0,0019056442421398975
year	0,001952302492670381
sector	0,0015701109343741065
revenue	0,0011487680720973657
key	3,6081154869435975E-4

Bitte achten Sie hierbei auf die Wörter „market“ und „report“. Bei jedem Anstieg der Dokumentenanzahl im Gesamtkorpus sinken hierbei die TF-IDF Werte und bekräftigen die Aussage, dass diese Wörter sehr oft in Marktstudien verwendet werden und somit in diesem Sinne auch zur Gruppe der Stoppwörter zählen. Bei den Wörtern „metal“ und „food“ hingegen zeichnet sich bei jedem Anstieg der Dokumentenanzahl ein signifikanter Anstieg des TF-IDF Wertes ab.

Anhang E CD

Die beigelegte CD enthält die folgenden Daten:

- Komplette Kategorienontologie
- Ausgewählte Links zu den Lernmaterialien aus reports-research.com
- Java Quellcode der Anwendung
- Der Gesamtkorpus als XML Datei
- Ausgewählte Kategorienwortliste
- Cue Dictionary
- Datei zur englischen Stoppwortliste
- Datei zur Abkürzungsliste

„Ich versichere, die von mir vorgelegte Arbeit selbständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen teilen noch keiner anderen Prüfungsbehörde vorgelegen."

Köln, den 7. Juli 2010